

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SCÉNARIOS DE TRI POUR PERMUTATIONS SIGNÉES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

JACQUELINE RWIRANGIRA

MAI 2008

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

*À ma merveilleuse petite Maya*

*À mon bel amour André*

## REMERCIEMENTS

Mes premiers remerciements s'adressent tout naturellement à ma directrice de recherche Anne Bergeron pour sa patience, sa disponibilité et sa générosité. Je suis très touchée par tout le temps qu'elle a mis à lire, relire et corriger ce travail. J'ai particulièrement apprécié sa façon simple et élégante de présenter des concepts assez complexes qui les rend accessibles pour une débutante comme moi. Je m'estime privilégiée d'avoir pu travailler sous sa direction.

J'ai commencé à travailler sur le sujet de mon mémoire durant le projet Genoc au Laboratoire de Génomique Comparée. Je voudrais remercier mes collègues de ce projet André Levasseur, Olivier Gingras, Yannick Gingras et les professeurs Cédric Chauve et Anne Bergeron. Les discussions que j'ai eues avec chacun d'eux m'ont aidée à la réalisation de ce projet.

Je tiens à remercier mes parents, ma soeur, mes frères et mes amies de longue date, Rosette et Odette, pour leurs encouragements, leur présence constante et leur amour inconditionnel.

Je ne trouve pas les mots pour exprimer ma gratitude à André. Durant toute cette période, il m'a soutenue, encouragée, conseillée et surtout il m'a aimée. Quel bonheur de l'avoir dans ma vie.

Pour terminer, mes pensées vont à ma fille, Maya, rien de mieux que le sourire de mon petit trésor pour me rappeler ce qui compte vraiment.

## TABLE DES MATIÈRES

TABLE DES FIGURES . . . . .	viii
INTRODUCTION . . . . .	1
CHAPITRE I	
SCÉNARIOS D'ÉVOLUTION ET INTERVALLES COMMUNS . . . . .	3
1.1 Notions préliminaires . . . . .	5
1.1.1 Les permutations signées . . . . .	6
1.1.2 Les intervalles . . . . .	6
1.1.3 Les inversions . . . . .	7
1.1.4 Un scénario entre deux permutations . . . . .	7
1.2 Les intervalles communs . . . . .	11
1.2.1 Qu'est-ce qu'un <i>intervalle commun</i> ? . . . . .	11
1.2.2 Le comportement des intervalles communs durant un scénario . . . . .	12
1.3 Algorithmes pour identifier les intervalles communs . . . . .	14
1.3.1 L'algorithme de Bergeron et al. . . . .	15
1.4 Différents types de scénarios . . . . .	16
1.4.1 Les scénarios parfaits . . . . .	17
1.4.2 Les scénarios parcimonieux . . . . .	17
1.4.3 Les scénarios parfaits et parcimonieux . . . . .	18
1.4.4 Les scénario parfaits les plus courts . . . . .	19
CHAPITRE II	
DES INTERVALLES COMMUNS AUX INTERVALLES FORTS . . . . .	22
2.1 Intervalles forts . . . . .	22
2.1.1 Les intervalles forts sont moins nombreux que les intervalles communs	24
2.1.2 Inverser un intervalle fort ne brise jamais un intervalle commun d'une permutation . . . . .	25
2.2 Arbre d'inclusion des intervalles forts . . . . .	26
2.2.1 Présentation de l'arbre d'inclusion des intervalles forts . . . . .	26

2.2.2	Les propriétés découlant de l'arbre d'inclusion . . . . .	28
2.3	L'arbre d'inclusion des intervalles forts avec les feuilles ordonnées . . . . .	30
2.3.1	Ordonner les feuilles d'un arbre d'inclusion des intervalles forts . . . . .	30
2.3.2	Les conséquences de l'ordonnement des feuilles d'un arbre d'inclusion des intervalles forts . . . . .	31
2.3.3	La permutation-quotient . . . . .	33
2.4	La classification des noeuds d'un arbre d'inclusion . . . . .	37
2.5	Les arbres-PQ . . . . .	40
2.5.1	Introduction aux arbres-PQ . . . . .	41
2.5.2	Quelques définitions de base pour comprendre les arbres PQ . . . . .	41
2.5.3	Des arbres d'inclusion aux arbres PQ . . . . .	42
CHAPITRE III		
IDENTIFIER UN SCÉNARIO PARFAIT ET PARCIMONIEUX . . . . .		47
3.1	Les arbres des intervalles forts . . . . .	47
3.1.1	Accorder les signes aux noeuds de l'arbre . . . . .	47
3.1.2	Différents types d'arbres . . . . .	51
3.1.3	La présentation des données . . . . .	53
3.2	Les scénarios parfaits et parcimonieux . . . . .	55
3.2.1	Identifier un scénario parfait et parcimonieux avec un arbre des intervalles forts défini . . . . .	61
3.2.2	Un scénario parfait et parcimonieux avec un arbre des intervalles forts non-ambigu . . . . .	64
3.2.3	Un scénario parfait et parcimonieux avec un arbre des intervalles forts ambigu . . . . .	68
3.3	La complexité des algorithmes de calcul des scénarios parfaits et parcimonieux . . . . .	72
3.3.1	La complexité de l'algorithme de calcul d'un scénario parfait et parcimonieux avec un arbre défini . . . . .	74
3.3.2	La complexité de l'algorithme de calcul d'un scénario parfait et parcimonieux avec un arbre non-ambigu . . . . .	75
3.3.3	La complexité de l'algorithme de calcul d'un scénario parfait et parcimonieux avec un arbre ambigu . . . . .	77
CONCLUSION . . . . .		78

ANNEXE A	
LA DISTANCE D'INVERSION . . . . .	80
A.1 Le cas de la permutation $P = (2 \bar{5} \bar{3} 1 4)$ . . . . .	83
A.2 Le cas de la permutation $P = (\bar{4} 6 1 \bar{3} \bar{5} \bar{2})$ . . . . .	84
BIBLIOGRAPHIE . . . . .	86

## Liste des Algorithmes

1	L'algorithme qui affecte un signe au noeud $I$ de l'arbre des intervalles forts passé en paramètre. . . . .	51
2	L'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est défini. . . . .	62
3	L'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est non-ambigu. . . . .	69
4	L'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est ambigu. . . . .	71
5	Calcul d'un scénario parfait et parcimonieux de $P$ . . . . .	75
6	Calcul de la borne inférieure de la distance d'inversion d'une permutation. . . . .	83



## TABLE DES FIGURES

1.1	L'ancien génome $C$ évolue pour donner les génomes actuels $A$ et $B$ . . .	4
1.2	Une autre représentation de l'évolution du génome $C$ vers $A$ et $B$ . . . .	5
1.3	Les deux scénarios par inversions, $S_1$ et $S_2$ , transforment $P$ en l'identité $Id_n$ . . . . .	8
1.4	La composition, $Q \circ P$ , des permutations $P$ et $Q$ . . . . .	9
1.5	La permutation inverse de $P$ notée $P^{-1}$ . . . . .	10
1.6	La définition de la permutation-quotient . . . . .	10
1.7	Un scénario qui trie $P$ vers l'identité . . . . .	13
1.8	Un scénario parfait qui trie $P$ vers l'identité . . . . .	17
1.9	Un scénario parcimonieux qui trie $P$ vers l'identité . . . . .	18
1.10	Un scénario parfait et parcimonieux qui trie $P$ vers l'identité . . . . .	19
1.11	Un scénario parcimonieux qui trie $P$ vers l'identité . . . . .	20
1.12	Le scénario parfait le plus court qui transforme $P$ en l'identité renversée	20
1.13	La relation entre les différents types de scénarios . . . . .	21
2.1	Les intervalles communs de la permutation $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . . .	23
2.2	Les intervalles communs de $P = (4\ 2\ 3\ 1\ 11\ 10\ 9\ 8\ 7\ 6\ 5)$ . . . . .	24
2.3	Les intervalles forts de $P = (4\ 2\ 3\ 1\ 11\ 10\ 9\ 8\ 7\ 6\ 5)$ . . . . .	25

2.4	L'effet d'une inversion d'un intervalle qui chevauche un intervalle commun de $P$ sur le nombre d'intervalles communs de $P$ . . . . .	26
2.5	Arbre d'inclusion des intervalles forts de $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . . . . .	27
2.6	Dans un arbre d'inclusion, chaque noeud a au moins deux enfants . . . . .	28
2.7	Arbre d'inclusion des intervalles forts de $P$ avec un ordonnancement arbitraire de feuilles . . . . .	31
2.8	Arbre d'inclusion des intervalles forts d'une permutation avec les feuilles ordonnées selon cette permutation . . . . .	31
2.9	Arbre d'inclusion des intervalles forts d'une permutation avec des feuilles ordonnées selon la permutation identité . . . . .	32
2.10	Accorder les rangs aux enfants d'un noeud selon un ordre croissant . . . . .	34
2.11	Accorder les rangs aux enfants d'un noeud selon un ordre croissant . . . . .	35
2.12	Arbre d'inclusion des intervalles forts avec une permutation-quotient sur chaque noeud interne . . . . .	38
2.13	Un exemple d'un arbre-PQ . . . . .	43
2.14	Arbre-PQ des intervalles forts de la permutation $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . . . . .	44
3.1	Intuition sur le sens du tri d'un noeud dont le parent est linéaire . . . . .	49
3.2	Arbre des intervalles forts signé de $P = (\bar{1}\ \bar{4}\ 2\ 5\ 3\ 7\ \bar{8}\ 6\ 9)$ . . . . .	50
3.3	Arbre des intervalles forts de $P = (\bar{1}\ \bar{4}\ 2\ 5\ 3\ 7\ \bar{8}\ 6\ 9)$ est non-ambigu . . . . .	52
3.4	Arbre des intervalles forts de $P = (7\ 8\ \bar{9}\ 4\ 5\ \bar{6}\ 1\ 2\ \bar{3})$ est défini . . . . .	52
3.5	Arbre des intervalles forts de $P = (2\ 5\ 7\ 4\ 6\ 1\ 3)$ est ambigu . . . . .	53

3.6	Les permutations issues de la comparaison des chromosomes X de l'homme, de la souris et du rat. . . . .	54
3.7	L'arbre $T_f(P)$ issu de la comparaison des chromosomes X de la souris et du rat. . . . .	55
3.8	L'arbre $T_f(P)$ issu de la comparaison des chromosomes X de homme et du rat. . . . .	56
3.9	L'arbre $T_f(P)$ issu de la comparaison des chromosomes X de homme et de la souris. . . . .	57
3.10	L'intervalle $I$ inclus dans l'intervalle fort $J$ commute avec les intervalles communs de $P$ disjoints de $J$ . . . . .	58
3.11	L'intervalle $I$ commute avec tout intervalle commun de $P$ inclus dans un enfant de $J$ . . . . .	58
3.12	Tous les intervalles forts inclus dans $I$ doivent avoir le même parent. . .	59
3.13	Si $I$ est l'union des enfants de $J$ et $I \neq J$ , alors $I$ chevauche l'intervalle formé par le restant de $J$ et un des intervalles qui forme $I$ . . . . .	60
3.14	Un scénario qui trie le chromosome X du rat vers celui de la souris . . .	63
3.15	Un scénario qui trie le chromosome X du rat vers celui de la souris . . .	64
3.16	Ce scénario trie le chromosome X du rat vers celui de l'homme : étape $A$	67
3.17	Ce scénario trie le chromosome X du rat vers celui de l'homme : étape $B$	68
3.18	Un scénario qui trie la permutation $(\overline{4} \ 6 \ 1 \ \overline{3} \ \overline{5} \ \overline{2})$ vers $\overline{1d}$ : étape $A$ . . .	72
3.19	Un scénario qui trie la permutation $(\overline{4} \ 6 \ 1 \ \overline{3} \ \overline{5} \ \overline{2})$ vers $\overline{1d}$ : étape $B$ . . .	73
3.20	Un exemple d'une permutation difficile à trier . . . . .	73

3.21	Deux scénarios qui trient la permutation $(2\ 4\ 1\ 3)$ , l'un vers $Id$ et l'autre vers $\overline{Id}$ . Nous gardons le scénario vers $\overline{Id}$ car il est le plus court . . . . .	74
A.1	Exemples d'intervalles élémentaires de la permutation $(0\ \overline{4}\ 6\ 1\ \overline{3}\ \overline{5}\ \overline{2}\ 7)$ .	81
A.2	La trace qui montre les cycles de $P = (2\ \overline{5}\ \overline{3}\ 1\ 4)$ . . . . .	82
A.3	Un scénario de longueur 5 qui trie $(2\ \overline{5}\ \overline{3}\ 1\ 4)$ vers $\overline{Id}$ est parcimonieux.	84
A.4	Un scénario de longueur 6 qui trie $(\overline{4}\ 6\ 1\ \overline{3}\ \overline{5}\ \overline{2})$ vers $Id$ est parcimonieux.	85
A.5	Un scénario de longueur 6 qui trie $(\overline{4}\ 6\ 1\ \overline{3}\ \overline{5}\ \overline{2})$ vers $\overline{Id}$ est parcimonieux.	85

## DÉCLARATION DES VARIABLES

- $A$  Un génome actuel quelconque.
- $b$  Un point dans une permutation.
- $B$  Un génome actuel quelconque différent de  $[A]$ .
- $c$  Le nombre de cycles dans une permutation.
- $C$  Un génome ancien quelconque.
- $d(P)$  La distance d'inversion d'une permutation  $P$ .
- $f$  Une feuille quelconque d'un arbre.
- $h$  Un élément quelconque d'une permutation  $P$ .
- $h'$  Un autre élément quelconque de  $P$  différent de  $[h]$ .
- $H$  Un intervalle d'une permutation différent des intervalles  $[I]$  et  $[J]$ .
- $I$  Un intervalle d'une permutation  $[P]$  ou un noeud de son arbre des intervalles forts.
- $Id_n$  La permutation identité comprenant  $[n]$  éléments.
- $\overline{Id_n}$  La permutation identité inversée qui comprend  $n$  éléments.
- $J$  Un autre intervalle de la permutation  $[P]$  différent de l'intervalle  $[I]$ .
- $K$  Un intervalle d'une permutation.
- $K'$  Un autre intervalle d'une permutation différente de  $[K]$ .
- $k$  Le nombre d'intervalles forts inclus dans un intervalle donné.
- $\ell$  Un élément quelconque d'une permutation  $[P]$ .
- $m$  Le nombre de noeuds dans un arbre d'inclusion.
- $n$  Le nombre d'éléments d'une permutation.
- $P$  Une permutation sur  $[n]$  éléments.
- $P'$  Une autre permutation sur  $[n]$  éléments différente de  $[P]$ .
- $P^{-1}$  La permutation inverse de  $[P]$ .

- $P[i]$  L'élément situé à la position  $i$  dans la permutation  $[P]$ .
- $P_I$  La permutation-quotient associée au noeud  $[I]$ .
- $Q$  Une permutation sur  $[n]$  éléments différente de  $[P]$ .
- $S$  Un scénario quelconque.
- $S_1$  Un scénario d'une permutation  $[P]$  vers l'identité.
- $S_2$  Un scénario de la même permutation  $[P]$  vers l'identité mais différent de  $[S_1]$ .
- $T$  Un arbre d'inclusion d'une permutation  $[P]$ .
- $T'$  Un autre arbre d'inclusion de la permutation  $[P]$  qui est différent de  $[T]$ .
- $T_f(P)$  L'arbre des intervalles forts de la permutation  $[P]$ .
- $x$  Un autre élément d'une permutation qui est différent de  $[\ell]$ .
- $y$  Un élément quelconque d'une permutation  $[P]$ .
- $y'$  Un autre élément quelconque de  $[P]$  différent de  $[y]$ .
- $\Sigma$  Un ensemble d'entiers.
- $\Phi$  Un ensemble de sous-ensembles de  $[\Sigma]$  ou une partition associé à l'ensemble  $[\Sigma]$ .

## RÉSUMÉ

En bioinformatique, un des problèmes très étudiés est la reconstitution des événements évolutifs qui transforment un génome  $A$  en un autre génome  $B$ . L'ordre des gènes dans les génomes est souvent modélisé par des permutations signées.

Dans ce travail, nous définissons des liens entre le problème de tri des permutations signées par inversions et la conservation des structures combinatoires communes aux génomes à comparer.

En utilisant les arbres des intervalles forts (Bergeron et al. (3)), nous démontrons que même si le calcul d'un scénario parfait et parcimonieux est difficile (Figeac et Varré (11)), il peut se faire d'une façon efficace pour une grande classe de permutations.

Nous avons appliqué ces résultats à la comparaison des chromosomes X de l'humain, de la souris et du rat, basée sur les données de l'article de Gibbs et al. (12).

### MOTS CLÉS :

Genomique comparee

Scénarios d'évolution

Tri par inversions

Intervalles communs

## INTRODUCTION

La reconstruction de scénario d'évolution basée sur les opérations de réarrangement entre les génomes, telle que l'inversion, est un outil puissant dans la compréhension de l'évolution des espèces. Par exemple, ces dernières années, plusieurs scénarios d'évolution entre les génomes des mammifères ont été proposés dans les articles de Bourque et al. (8) et de Gibbs et al. (12) en utilisant les logiciels tels que GRIMM et MGR décrits dans les articles de Bourque et Pevzner (7) et Tesler (21).

Le calcul de tels scénarios est basé sur le problème de tri des permutations signées par inversions : étant donné deux génomes modélisés par deux permutations sur les mêmes éléments, trouver la "bonne" séquence d'inversions qui transforme l'un des deux génomes en l'autre. Ce problème a été introduit par Sankoff (19).

Dans l'approche classique, une bonne séquence d'inversions qui transforme l'une permutation en l'autre est une séquence parcimonieuse. Le problème de calcul d'une séquence d'inversions parcimonieuse qui transforme une permutation signée en une autre est bien maîtrisé et il se résout en temps polynomial. Les auteurs Hannenhalli et Pevzner (14) ont développé une théorie qui est à la base de plusieurs algorithmes qui résolvent ce problème. Parmi eux, citons l'algorithme de Tannier et al. (20) qui possède la meilleure complexité théorique connue présentement.

Un des problèmes soulevés par l'utilisation de séquences d'inversions parcimonieuses est qu'elles brisent, parfois, les structures combinatoires communes aux espèces à comparer. Or, les biologistes croient que si deux génomes voisins génétiquement possèdent une structure commune, alors il est probable que leur ancêtre commun possède la même structure et qu'elle a été conservée au cours de l'évolution pour des raisons fonctionnelles, donc, de telles structures devraient être préservées.



Par conséquent, il devient intéressant de trouver des séquences d'inversions de longueur minimale qui transforment l'un des deux génomes en l'autre mais, sans briser leurs structures communes. Nous appelons une telle séquence, *un scénario parfait et parcimonieux*. Cette méthode peut être vue comme une variante de l'approche classique où le critère de parcimonie a été relâché dans le but d'inclure la conservation des structures communes. D'après Figeac et Varré (11), ce problème est NP-difficile.

Dans l'article de Bérard et al. (2), les auteurs démontrent qu'il existe une grande classe de permutations sur laquelle le calcul d'un scénario parfait et parcimonieux qui transforme une permutation signée en une autre se fait en temps polynomial. Ils utilisent une structure de données particulière appelée l'arbre des intervalles forts qui a été introduit dans l'article de Bergeron et al. (3).

Le présent travail est basé sur l'article de Bérard et al. (2). Nous faisons une présentation simple et complète des résultats de l'article de Bérard et al. (2) accompagnée d'exemples simples et faciles à comprendre. Ensuite, nous proposons une implantation des algorithmes de calcul d'un scénario parfait et parcimonieux donnés dans cet article. Noter que, pour tous les résultats tirés de cet article, nous ne donnerons pas des références dans la suite du travail.

Dans le premier chapitre, nous définissons les concepts de base de permutations, inversions, scénarios, intervalles communs, etc. Dans le deuxième chapitre, nous introduisons la notion d'intervalles forts et de leur représentation sous forme d'un arbre. Dans le troisième et dernier chapitre, nous discutons des algorithmes qui identifient un scénario parfait et parcimonieux d'une permutation signée à partir de l'arbre d'intervalles forts de cette permutation.

## CHAPITRE I

### SCÉNARIOS D'ÉVOLUTION ET INTERVALLES COMMUNS

Durant l'évolution, les génomes subissent non seulement des mutations au niveau de leur séquence, mais aussi des mutations à grande échelle appelées les *réarrangements génomiques*. Ce sont des déplacements de gènes ou de fragments de chromosomes au niveau du génome entier.

Il existe plusieurs sortes de réarrangements. Certains s'opèrent à l'intérieur du même chromosome, ce sont les réarrangements intra-chromosomiques et d'autres entre plusieurs chromosomes, ce sont les réarrangements inter-chromosomiques.

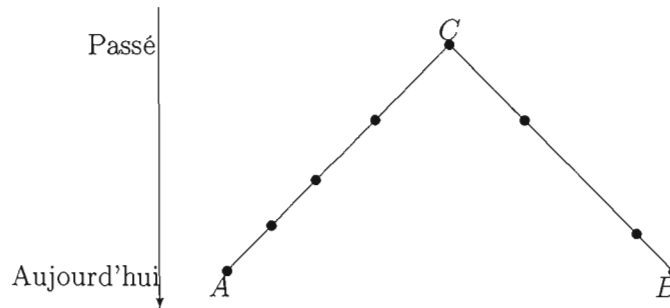
Nous nous intéresserons à une opération de réarrangement particulière, nommée l'*inversion*, pour deux raisons : d'abord, parce qu'elle est l'opération de réarrangement la plus documentée et aussi, parce que c'est un événement d'évolution fréquent surtout chez les chloroplastes, les mitochondries et les bactéries.

Une inversion est un réarrangement intra-chromosomique qui consiste en une cassure d'un fragment de chromosome, suivi d'une réparation qui modifie l'orientation du fragment.

Il arrive souvent qu'un ensemble de gènes voisins dans un génome se retrouve dans plusieurs autres génomes à l'ordre et l'orientation près. En biologie, on croit que cette conservation de structures combinatoires communes dans plusieurs génomes est justifiée par une pression fonctionnelle qui requiert que ces gènes restent voisins.

Par conséquent, il est raisonnable de penser qu'elles aient toujours été présentes au cours du processus d'évolution, et qu'elles devraient le rester dans les scénarios d'évolution inférés.

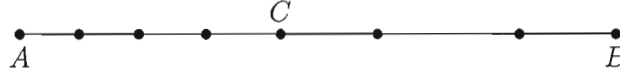
La recherche d'un *scénario d'évolution* entre les génomes  $A$  et  $B$  consiste à trouver une séquence de réarrangements que nous pouvons faire subir au génome  $A$  pour avoir le génome  $B$ . Or, en biologie, si deux génomes  $A$  et  $B$  sont phylogénétiquement proches, alors il y a eu un génome  $C$ , dans le passé, qui est l'ancêtre de  $A$  et  $B$ . Ensuite,  $C$  a subi deux séquences de réarrangements distinctes. Une de ces séquences l'a transformé en génome  $A$  et l'autre en génome  $B$  tel qu'illustré dans l'arbre de la Figure 1.1.



**FIG. 1.1** Dans cet arbre phylogénétique, le génome  $C$  a évolué pour donner les génomes actuels  $A$  et  $B$ . Les points noirs représentent le nouvel état du génome après une opération de réarrangement. À la fin, le génome  $C$  est transformé en génome  $A$  sur une des arêtes et en génome  $B$  sur l'autre arête.

L'arbre de la Figure 1.1 peut aussi être représenté sous forme d'une ligne comme le montre la Figure 1.2, si on s'imagine qu'on tire sur les deux feuilles  $A$  et  $B$  de l'arbre. Avec la représentation de la Figure 1.2, nous remarquons la similitude avec le problème de trouver un scénario qui transforme le génome  $A$  en génome  $B$ .

Ce qui explique le lien entre un arbre phylogénétique qui représente un ancien génome  $C$  (l'ancêtre) qui évolue vers les génomes actuels  $A$  et  $B$  et un scénario d'évolution entre  $A$  et  $B$ .



**FIG. 1.2** Une représentation différente de l'évolution du génome  $C$  vers les génomes actuels  $A$  et  $B$ . Les informations contenues dans cette figure sont exactement les mêmes que celles de la Figure 1.1, seule la forme a changé d'un arbre à une ligne.

Notons que, dans le cas d'un scénario de  $A$  vers  $B$ , l'ancêtre commun  $C$  est représenté par un des états intermédiaires du génome dans le scénario.

Si  $A$  et  $B$  partagent un trait commun que  $C$  ne possédait pas, cela implique que pendant l'évolution,  $C$  a subi deux modifications indépendantes qui ont eu exactement la même conséquence sur les génomes actuels  $A$  et  $B$ . Cette hypothèse est moins plausible que celle selon laquelle l'ancêtre  $C$  possédait un trait qui a été préservé chez ses deux descendants  $A$  et  $B$ . C'est pour cette raison que nous allons nous intéresser aux scénarios qui préservent les structures combinatoires communes aux génomes comparés.

Dans le présent travail, nous allons modéliser un groupe de gènes communs à plusieurs génomes par des entiers et les génomes seront modélisés par des permutations signées sur l'ensemble de ces entiers. Les éléments de la permutation représentent les gènes ou les segments de génomes identiques, tandis que le signe de l'élément modélise l'orientation du brin sur lequel se trouve le gène correspondant.

Les structures communes seront, quant à elles, modélisées par les *intervalles communs* entre les permutations qui représentent les génomes. Les intervalles communs seront étudiés plus profondément dans ce chapitre.

## 1.1 Notions préliminaires

Dans cette section, nous définissons les concepts de base nécessaires pour comprendre la notion d'intervalles communs. Pour chaque notion définie, nous donnons un exemple pour l'illustrer.

### 1.1.1 Les permutations signées

Une *permutation signée* sur  $n$  éléments est une permutation sur l'ensemble d'entiers  $\{1, \dots, n\}$  dans laquelle tout élément a un signe soit positif, soit négatif. Chaque élément de l'ensemble doit être présent dans la permutation une et une seule fois et la permutation ne contient que les éléments de l'ensemble. Pour représenter les entiers négatifs dans la permutation, nous plaçons une barre horizontale au-dessus de l'élément.

Prenons l'exemple de l'ensemble  $\Sigma = \{1, \dots, 8\}$ , les permutations signées sur  $\Sigma$   $P_1 = (\bar{2} \bar{1} 4 3 5 \bar{8} 6 7)$  et  $P_2 = (1 2 \bar{5} 3 7 \bar{6} 8 4)$  sont valides. Par contre, les permutations  $P_3 = (\bar{2} \bar{1} 7 3 5 \bar{8} 6 \bar{7} 4)$  et  $P_4 = (\bar{2} \bar{1} \bar{4} 3 \bar{8} 6 7)$  sont invalides car  $P_3$  contient une répétition de l'élément 7 avec signes opposés, tandis que  $P_4$  ne contient pas l'élément 5.

Une permutation signée sur  $n$  éléments est dite *identité* notée  $Id_n$  si elle est de la forme  $(1 2 \dots n)$  et elle est dite *identité inversée* notée  $\overline{Id_n}$  si elle est de la forme  $(\bar{n} \dots \bar{2} \bar{1})$ .

### 1.1.2 Les intervalles

Un *intervalle* d'une permutation signée est un segment d'éléments consécutifs dans la permutation. On peut définir l'intervalle en donnant l'ensemble de ses éléments non signés appelé le *contenu* de l'intervalle. Cependant, tout ensemble d'entiers ne correspond pas nécessairement à un intervalle d'une permutation donnée.

On peut aussi identifier un intervalle par ses bornes gauche et droite. Les bornes sont mises entre parenthèses lorsque la permutation est l'identité et entre crochets sinon. Ainsi, si on a une permutation  $P = (p_1, \dots, p_i, \dots, p_j, \dots, p_n)$ , l'intervalle qui contient les éléments  $\{p_i, \dots, p_j\}$  peut aussi s'écrire  $(i..j)$  si  $P$  est l'identité, et  $[i..j]$  si  $P$  n'est pas l'identité.

Prenons l'exemple de la permutation  $P = (\bar{2} \bar{1} 4 3 \bar{8} 5 6 7)$ . Les ensembles  $\{1, 2\}$ ,  $\{1, 2, 4\}$  et  $\{3, 4, 8\}$  sont des exemples d'intervalles de  $P$ . Chaque intervalle est noté par l'ensemble des éléments non signés qu'il contient, soit son contenu. On peut aussi utiliser

l'autre notation. Ainsi, l'intervalle  $\{3, 4, 8\}$ , par exemple, peut aussi s'écrire  $[3..5]$  car la permutation  $P$  n'est pas l'identité.

### 1.1.3 Les inversions

Une *inversion* d'un intervalle d'une permutation signée inverse l'ordre des éléments de l'intervalle, tout en changeant leur signe. Le reste de la permutation est inchangé. Noter qu'à toute inversion correspond un intervalle de la permutation sur laquelle elle est effectuée.

Ceci nous amène à traiter les inversions comme des intervalles, et à représenter une inversion par l'intervalle correspondant. Si  $P$  est une permutation, nous notons  $\overline{P}$  la permutation obtenue en inversant au complet la permutation  $P$ .

Prenons l'exemple de la permutation signée  $P = (\overline{2} \ \overline{1} \ 4 \ 3 \ 5 \ \overline{8} \ 6 \ 7)$  sur l'ensemble  $\{1, \dots, 8\}$ . L'inversion de l'intervalle  $\{1, 3, 4\}$  donne la permutation  $(\overline{2} \ \overline{3} \ \overline{4} \ 1 \ 5 \ \overline{8} \ 6 \ 7)$ , tandis que  $\overline{P} = (\overline{7} \ \overline{6} \ 8 \ \overline{5} \ \overline{3} \ \overline{4} \ 1 \ 2)$ .

### 1.1.4 Un scénario entre deux permutations

**Définition 1.** Soient  $P$  et  $Q$  deux permutations signées sur  $n$  éléments. Un *scénario* entre  $P$  et  $Q$  est une séquence d'inversions distinctes qui transforment la permutation  $P$  en la permutation  $Q$  ou en  $\overline{Q}$ . La *longueur* d'un scénario est le nombre d'inversions qu'il contient. Lorsque  $Q$  est la permutation identité, un scénario de  $P$  vers  $Q$  est simplement appelé un scénario de  $P$ .

Prenons l'exemple de  $P = (1 \ \overline{3} \ \overline{5} \ \overline{2} \ 4 \ 6)$  de la Figure 1.3. Nous avons deux séquences d'inversions qui transforment  $P$  en l'identité. Chaque scénario,  $S_1$  et  $S_2$ , occupe une moitié de la figure. Chaque inversion est marquée dans la colonne de gauche de sa moitié et la permutation résultante correspondante dans la colonne de droite.

Ces scénarios sont appelés des scénarios de  $P$  étant donné qu'ils transforment  $P$  en l'identité. Le scénario  $S_1$  est de longueur 5 et le scénario  $S_2$  est de longueur 3.

inversion $\Rightarrow$ permutation résultante	inversion $\Rightarrow$ permutation résultante
$P = (1 \ 3 \ 5 \ 2 \ 4 \ 6)$	$P = (1 \ 3 \ 5 \ 2 \ 4 \ 6)$
$\{2, 3, 5\} \Rightarrow (1 \ 2 \ 5 \ 3 \ 4 \ 6)$	$\{2, 4, 5\} \Rightarrow (1 \ 3 \ 4 \ 2 \ 5 \ 6)$
$\{3, 5\} \Rightarrow (1 \ 2 \ 3 \ 5 \ 4 \ 6)$	$\{2, 4\} \Rightarrow (1 \ 3 \ 2 \ 4 \ 5 \ 6)$
$\{3\} \Rightarrow (1 \ 2 \ 3 \ 5 \ 4 \ 6)$	$\{2, 3\} \Rightarrow (1 \ 2 \ 3 \ 4 \ 5 \ 6)$
$\{4\} \Rightarrow (1 \ 2 \ 3 \ 5 \ 4 \ 6)$	
$\{4, 5\} \Rightarrow (1 \ 2 \ 3 \ 4 \ 5 \ 6)$	
$S_1$	$S_2$

**FIG. 1.3** Les deux scénarios,  $S_1$  et  $S_2$  transforment  $P$  en l'identité. Nous pouvons les appeler les scénarios de  $P$  car la permutation cible est l'identité dans les deux cas. Il existe plusieurs autres scénarios de  $P$ , de longueurs variées.

**Définition 2.** Soit deux permutations sur  $n$  éléments  $P$  et  $Q$ . Appliquer la composition de  $P$  et  $Q$ , notée  $Q \circ P$ , consiste à appliquer la permutation  $Q$  sur la permutation résultante à l'application de la permutation  $P$ .

Prenons, par exemple, deux permutations sur  $n$  éléments, soient  $P = (\bar{2} \ 1 \ 4 \ 3 \ 5 \ \bar{8} \ 6 \ 7)$  et  $Q = (3 \ \bar{1} \ \bar{6} \ \bar{7} \ 2 \ 4 \ 5 \ 8)$ . Nous allons décrire la façon d'identifier les éléments de la permutation issue de la composition de ces deux permutations, notée  $Q \circ P$ .

Mais, avant, mentionnons qu'il existe une règle qui dit que  $P(-a) = -P(a)$ . En effet, sachant que  $P(a)$  veut dire l'élément de  $P$  à la position  $a$ , si la position fournie est négative, soit  $-a$ , alors on retourne l'élément à la position  $a$  mais avec un signe opposé.

Par exemple, soit une permutation  $P = (\bar{2} \ 1 \ 4 \ 3 \ 5 \ \bar{8} \ 6 \ 7)$ . Pour identifier l'élément  $P(-3)$ , on identifie l'élément à la position 3 de la permutation  $P$ , soit 4, puis on inverse son signe, ce qui donne  $-4$ . Cette règle est nécessaire étant donné que nous travaillons avec des permutations signées.

Maintenant, par la définition, nous savons que, pour tout entier  $i \in \{1, \dots, n\}$ , on a  $Q \circ P(i) = Q(P(i))$ . Ainsi, dans notre exemple, pour  $i = 1$ , nous avons  $P(1) = -2$ , donc  $Q(P(1)) = Q(-2)$ . Or selon la règle ci-dessus, nous savons que  $Q(-2) = -Q(2)$ . Donc, sachant que  $Q(2) = -1$  on a  $Q \circ P(1) = -(-1) = 1$ , tandis que, pour  $i = 2$ , on a  $P(2) = -1$ , ce qui implique que  $Q(P(2)) = Q(-1)$ . Or, on sait que  $Q(-1) = -Q(1)$ , et comme  $Q(1) = 3$ , on a donc  $Q \circ P(2) = -(3) = -3$ . On continue avec  $i = 3$  et ainsi de suite jusqu'à  $i = n$  et la permutation  $Q \circ P$  est identifiée.

Dans la Figure 1.4 , nous y présentons les permutations  $P$  et  $Q$  ainsi que la permutation  $Q \circ P$  composée de ces deux permutations.

Position	1	2	3	4	5	6	7	8
$P$	$\bar{2}$	$\bar{1}$	4	3	5	$\bar{8}$	6	7
$Q$	3	$\bar{1}$	$\bar{6}$	$\bar{7}$	2	4	5	8
$Q \circ P$	1	$\bar{3}$	$\bar{7}$	$\bar{6}$	2	$\bar{8}$	4	5

**FIG. 1.4** Soient deux permutations  $P = (\bar{2} \ \bar{1} \ 4 \ 3 \ 5 \ \bar{8} \ 6 \ 7)$  et  $Q = (3 \ \bar{1} \ \bar{6} \ \bar{7} \ 2 \ 4 \ 5 \ 8)$ . La composition de ces deux permutations, notée  $Q \circ P$ , est égale à  $(1 \ \bar{3} \ \bar{7} \ \bar{6} \ 2 \ \bar{8} \ 4 \ 5)$ .

**Définition 3.** Soit une permutation  $P$  sur  $n$  éléments. La permutation *inverse* de  $P$ , notée  $P^{-1}$ , est une permutation sur  $n$  éléments telle que, si  $P$  possède l'élément  $i$  à la position  $j$ , alors  $P^{-1}$  aura l'élément  $j$  à la position  $i$ . Ainsi, nous avons  $P^{-1} \circ P = P \circ P^{-1} = Id$ .

Dans la Figure 1.5, nous y représentons l'exemple de la permutation  $P = (\bar{2} \ 5 \ 1 \ 6 \ 7 \ \bar{3} \ \bar{4} \ 8)$  et sa permutation inverse  $P^{-1} = (3 \ \bar{1} \ \bar{6} \ \bar{7} \ 2 \ 4 \ 5 \ 8)$ .

Pour simplifier les notations, nous allons toujours supposer que la permutation cible des scénarios est la permutation identité. Cette hypothèse n'entraîne aucune perte de généralité car il suffit d'effectuer une composition de permutations pour retrouver le cas général de deux permutations quelconques.

En effet, si on cherche un scénario entre deux permutations quelconques  $P$  et  $Q$ ,



	1	2	3	4	5	6	7	8
$P$	$\bar{2}$	5	1	6	7	$\bar{3}$	$\bar{4}$	8
$P^{-1}$	3	$\bar{1}$	$\bar{6}$	$\bar{7}$	2	4	5	8

FIG. 1.5 La permutation  $P = (\bar{2} \ 5 \ 1 \ 6 \ 7 \ \bar{3} \ \bar{4} \ 8)$  et sa permutation inverse.

on peut faire une composition des permutations  $P$  et  $Q$  avec la permutation inverse de l'une d'elles, par exemple  $Q$ . Ainsi, en appliquant la séquence d'inversions qui transforme  $Q^{-1} \circ P$  dans la permutation  $Q^{-1} \circ Q$  sur la permutation originale  $P$ , on obtient la permutation  $Q$ . Un exemple est donné dans la Figure 1.6.

$$\begin{array}{ll}
P = (\bar{2} \ \bar{1} \ 4 \ 3 \ 5 \ \bar{8} \ \bar{6} \ 7) & Q^{-1} \circ P = (1 \ \bar{3} \ \bar{7} \ \bar{6} \ 2 \ \bar{8} \ \bar{4} \ 5) \\
(\bar{2} \ \bar{1} \ 4 \ 3 \ 5 \ \bar{7} \ \bar{6} \ 8) & (1 \ \bar{3} \ \bar{7} \ \bar{6} \ 2 \ \bar{5} \ \bar{4} \ 8) \\
(\bar{2} \ \bar{1} \ \bar{5} \ \bar{3} \ \bar{4} \ \bar{7} \ \bar{6} \ 8) & (1 \ \bar{3} \ \bar{2} \ \bar{6} \ 7 \ \bar{5} \ \bar{4} \ 8) \\
(\bar{2} \ \bar{1} \ \bar{5} \ 6 \ 7 \ 4 \ 3 \ 8) & (1 \ \bar{3} \ \bar{2} \ 4 \ 5 \ \bar{7} \ \bar{6} \ 8) \\
(\bar{2} \ 5 \ 1 \ 6 \ 7 \ \bar{4} \ \bar{3} \ 8) & (1 \ 2 \ 3 \ 4 \ 5 \ \bar{7} \ \bar{6} \ 8) \\
Q = (\bar{2} \ 5 \ 1 \ 6 \ 7 \ \bar{3} \ \bar{4} \ 8) & Q^{-1} \circ Q = (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8)
\end{array}$$

FIG. 1.6 La transformation de la permutation  $P = (\bar{2} \ \bar{1} \ 4 \ 3 \ 5 \ \bar{8} \ \bar{6} \ 7)$  en la permutation  $Q = (\bar{2} \ 5 \ 1 \ 6 \ 7 \ \bar{3} \ \bar{4} \ 8)$  est simulé par la transformation de la permutation  $Q^{-1} \circ P$  en la permutation  $Q^{-1} \circ Q$ , où  $Q^{-1} = (3 \ \bar{1} \ \bar{6} \ \bar{7} \ 2 \ 4 \ 5 \ 8)$ .

Désormais, sachant que la permutation cible d'un scénario  $S$  d'une permutation  $P$  est l'identité, nous dirons que  $S$  trie  $P$  au lieu de  $S$  transforme  $P$  en  $Id$ .

En génomique comparée, inverser un chromosome entier ne modifie pas sa structure, ce qui explique le fait que l'ensemble de scénarios qui transforment la permutation  $P$  en la permutation  $Q$  contient aussi les séquences d'inversions qui transforment  $P$  en  $\bar{Q}$ .

## 1.2 Les intervalles communs

La notion d'*intervalle commun* a été introduite par Uno et Yagiura (22) pour modéliser le fait qu'un groupe de gènes peuvent être réarrangé dans plusieurs génomes différents tout en restant voisins. En bioinformatique, les *intervalles communs* sont utilisés pour détecter des associations fonctionnelles possibles entre les gènes.

En biologie, on croit que lorsque plusieurs gènes se retrouvent voisins dans différents génomes, à l'ordre et l'orientation près, ils tendent à encoder pour des protéines qui interagissent.

Dans cette section, nous verrons la définition formelle d'un intervalle commun. Ensuite, nous donnerons un exemple qui montre le comportement des intervalles communs d'une permutation durant un scénario de cette permutation.

### 1.2.1 Qu'est-ce qu'un *intervalle commun* ?

**Définition 4.** Soit deux permutations signées  $P$  et  $Q$  sur  $n$  éléments. Un *intervalle commun* à  $P$  et  $Q$  est un ensemble d'un ou plusieurs entiers qui constituent un intervalle aussi bien dans la permutation  $P$  que dans la permutation  $Q$ .

Par exemple, si on a une permutation  $P = (1 \ 3 \ 2 \ 5 \ 4 \ 6)$ , les intervalles communs entre  $P$  et la permutation identité  $Id_6$  sont  $\{2,3\}$ ,  $\{1,2,3\}$ ,  $\{4,5\}$ ,  $\{4,5,6\}$ ,  $\{2,3,4,5\}$ ,  $\{2,3,4,5,6\}$ ,  $\{1,2,3,4,5\}$ ,  $\{1,2,3,4,5,6\}$ , plus les singletons  $\{1\}$ ,  $\{2\}$ ,  $\dots$ ,  $\{6\}$ .

L'intervalle  $\{2,3,5\}$  est un exemple d'un intervalle de  $P$  qui n'est pas commun, car ce n'est pas un intervalle dans la permutation identité. De la même manière,  $\{2,3,4\}$  est un intervalle dans l'identité, mais ce n'est pas un intervalle dans  $P$ .

En faisant l'hypothèse que l'une des permutations est l'identité, nous dirons abusivement, dans la suite, qu'un intervalle  $I$  est un intervalle commun de  $P$  signifiant que c'est un intervalle commun de la permutation  $P$  et la permutation identité  $Id_n$ .

Notons qu'un intervalle commun de  $P$  et  $Q$  est aussi un intervalle dans les permutations  $\overline{P}$  et  $\overline{Q}$  car, en inversant une permutation entière, chaque élément garde ses voisins.

Il est clair que les singletons et l'ensemble  $\{1, 2, \dots, n\}$  sont toujours des intervalles communs. On les appelle *les intervalles communs triviaux*.

### 1.2.2 Le comportement des intervalles communs durant un scénario

Dans l'exemple qui suit, nous trions une permutation  $P$  vers l'identité. Notre but est d'observer le comportement des intervalles communs de  $P$  durant un de ses scénarios. En effet, à chaque inversion d'un scénario, des intervalles sont défaits et d'autres sont créés. Ainsi, lors d'un tri, si nous voulons préserver une structure combinatoire, tel un intervalle commun, il faut rejeter certains scénarios.

Considérons la permutation<sup>1</sup> signée  $P = (\overline{4} \ \overline{3} \ \overline{2} \ 1 \ \overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \overline{11} \ 12 \ 5 \ 6 \ 7)$ . Il existe plusieurs scénarios de longueurs différentes qui trient la permutation  $P$  vers la permutation identité. Le scénario  $S$  représenté dans la Figure 1.7 en est un exemple.

Dans ce scénario, certains intervalles communs de  $P$  ne sont plus des intervalles dans les permutations intermédiaires. Par exemple, l'intervalle commun  $\{1, 2, 3, 4\}$  de  $P$  n'est plus un intervalle de la permutation de la ligne  $r_7$ . On dit que l'intervalle commun  $\{1, 2, 3, 4\}$  est *brisé* durant le scénario  $S$ . Il redevient un intervalle dans la dernière permutation  $r_{10}$ .

C'est également le cas de l'intervalle commun  $\{13, 14, 15, 16\}$ . Il est brisé à la permutation de la ligne  $r_4$ . L'élément 13 n'est plus voisins des autres mais l'intervalle commun  $\{14, 15, 16\}$  existe encore. À la ligne suivante,  $r_5$ , c'est l'élément 16 qui s'éloigne des autres et l'intervalle commun  $\{13, 14, 15\}$  réapparaît. À  $r_6$ , l'ensemble  $\{13, 14, 15, 16\}$  redevient un intervalle et il le reste jusqu'à la fin.

---

<sup>1</sup>Cette permutation est obtenue en comparant le chromosome X du rat au chromosome X de la souris.

$$\begin{aligned}
r_1 : & ( \underline{\bar{4} \ \bar{3} \ \bar{2} \ 1 \ \bar{13} \ \bar{15} \ 14 \ \bar{16} \ 8 \ 9 \ 10 \ \bar{11} \ 12 \ 5 \ 6 \ 7} ) \\
r_2 : & ( \bar{12} \ 11 \ \bar{10} \ \bar{9} \ \bar{8} \ 16 \ \underline{\bar{14}} \ 15 \ 13 \ \bar{1} \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 ) \\
r_3 : & ( \bar{12} \ 11 \ \underline{\bar{10} \ \bar{9} \ \bar{8} \ 16 \ 14 \ 15} \ 13 \ \bar{1} \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 ) \\
r_4 : & ( \bar{15} \ \bar{14} \ \underline{\bar{16} \ 8 \ 9 \ 10 \ \bar{11} \ 12 \ 13} \ \bar{1} \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 ) \\
r_5 : & ( \underline{\bar{15} \ \bar{14} \ \bar{13} \ \bar{12} \ 11 \ \bar{10} \ \bar{9} \ \bar{8}} \ 16 \ \bar{1} \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 ) \\
r_6 : & ( \underline{8 \ 9 \ 10 \ \bar{11} \ 12 \ 13 \ 14 \ 15 \ 16} \ \bar{1} \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 ) \\
r_7 : & ( 1 \ \underline{\bar{16} \ \bar{15} \ \bar{14} \ \bar{13} \ \bar{12} \ 11 \ \bar{10} \ \bar{9} \ \bar{8}} \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 ) \\
r_8 : & ( 1 \ \bar{7} \ \bar{6} \ \bar{5} \ \bar{4} \ \bar{3} \ \bar{2} \ 8 \ 9 \ 10 \ \underline{\bar{11}} \ 12 \ 13 \ 14 \ 15 \ 16 ) \\
r_9 : & ( 1 \ \underline{\bar{7} \ \bar{6} \ \bar{5} \ \bar{4} \ \bar{3} \ \bar{2}} \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 ) \\
r_{10} : & ( 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 )
\end{aligned}$$

**FIG. 1.7** Le scénario  $S$  qui trie  $P$  vers l'identité. Pour représenter l'intervalle qui sera renversé à chaque étape, nous le soulignons. Après chaque inversion, la permutation à la ligne suivante montre la résultante avec la prochaine inversion à effectuer s'il y a lieu.

Par contre, il y a d'autres intervalles communs qui ont été préservés tout au long du scénario. L'intervalle commun  $\{5, 6, 7\}$  en est un exemple, on le retrouve dans toutes les permutations intermédiaires avec les éléments dans le même ordre, contrairement à l'intervalle  $\{14, 15\}$  qui est aussi préservé dans toutes les permutations, mais dont l'ordre change souvent durant le scénario.

Sachant que, durant un scénario d'une permutation, ses intervalles sont modifiés continuellement, on constate à quel point transformer une permutation en une autre, en préservant leurs intervalles communs, peut être une tâche délicate. Pour sauvegarder ce genre d'intervalles, il est important d'abord de les identifier. Voilà pourquoi nous allons discuter des algorithmes qui identifient les intervalles communs d'une permutation dans la section suivante.

### 1.3 Algorithmes pour identifier les intervalles communs

Intuitivement, pour identifier les intervalles communs entre deux permutations, il suffit de comparer tous les intervalles de l'une des permutations avec tous les intervalles de l'autre. Or, le nombre d'intervalles d'une permutation ne dépend que de sa longueur.

Aussi, pour une permutation sur un ensemble de  $n$  éléments, on a  $n$  intervalles qui commencent par le premier élément de la permutation,  $n - 1$  intervalles qui commencent par le deuxième,  $n - 2$  intervalles qui commencent par le troisième,  $\dots$ , un intervalle qui commence par le dernier élément de la permutation. Ce qui donne au total  $n + (n - 1) + \dots + 1$  intervalles. Ainsi, le nombre d'intervalles de la permutation sur  $n$  éléments est de  $n(n + 1)/2$  intervalles.

Le nombre maximal d'intervalles communs entre deux permutations est atteint lorsque les deux permutations comparées sont égales. Dans ce cas, tous les intervalles de l'une sont les intervalles de l'autre. Donc, sachant qu'on compare deux permutations de  $n$  éléments, on aura  $O(n^2)$  intervalles communs au maximum.

En général, un génome est composé de milliers de gènes (environ 4000 gènes dans un génome de bactérie et plusieurs dizaines de milliers pour les mammifères, par exemple), on constate alors que  $O(n^2)$  intervalles communs, au pire cas, c'est un nombre trop élevé.

Plusieurs chercheurs ont déjà travaillé sur le problème de l'identification des intervalles communs. Dans la suite de cette section, nous présentons trois différentes équipes qui ont solutionné ce problème.

En 2000, Uno et Yagiura (22) proposent le premier algorithme efficace de calcul des intervalles communs entre deux permutations sur  $n$  éléments avec une complexité théorique de  $O(n + N)$  où  $N$  est le nombre d'intervalles communs entre les permutations comparées, donc possiblement quadratique car il y a  $O(n^2)$  intervalles communs au pire cas.

Puis, en 2001, en se basant sur le travail d'Uno et Yagiura, Heber et Stoye (15) donnent un autre algorithme qui, contrairement à l'algorithme d'Uno et Yagiura, calcule les intervalles communs entre  $K$  permutations (avec  $K \geq 2$ ). La complexité de leur algorithme est de  $O(Kn + N)$ .

Dans leur algorithme, les auteurs Heber et Stoye utilisent un sous ensemble d'intervalles communs dits *irréductibles*. Un intervalle commun est irréductible s'il ne représente pas une union de deux intervalles communs qui se chevauchent.

Le principal défaut des algorithmes d'Uno et Yagiura (22) et Heber et Stoye (15) est que, d'après les auteurs mêmes, ils sont très compliqués et qu'ils utilisent des structures de données très complexes. En effet, même si, en théorie, ce sont des algorithmes efficaces, il existe des algorithmes plus simples en pratique, de complexité  $O(n^2)$ , qui s'exécutent plus rapidement.

Nous nous attarderons sur l'algorithme proposé par une troisième équipe, soit Bergeron et al. (3). Leur algorithme est présenté dans la section suivante.

### 1.3.1 L'algorithme de Bergeron et al.

Bergeron et al. (3) offrent la solution la plus simple à l'heure actuelle pour calculer les intervalles communs d'un ensemble de  $K$  permutations sur  $n$  éléments. La complexité de leur algorithme est la même que celle des algorithmes présentés dans la section précédente, soit  $O(n + N)$ .

Cependant, ils utilisent des structures de données élémentaires et un algorithme très simple, ce qui rend l'implémentation facile. L'aspect innovateur de leurs travaux se retrouve dans l'utilisation des *générateurs*.

Un *générateur* est une structure de données composée d'une paire de vecteurs qui nous permet de générer des intervalles communs. En conservant juste ce générateur, nous pouvons engendrer tous les intervalles communs au besoin.

Ainsi, les  $O(n^2)$  intervalles communs seront représentés dans un espace linéaire, car la taille d'un générateur est de  $2n$ . En effet, chacun des vecteurs qui le composent occupe un espace linéaire.

Ceci est une amélioration importante surtout lorsqu'on sait que la taille des données en biologie est très élevée. En effet, prenons, comme exemple, deux espèces voisines, soient l'homme et le chimpanzé. D'après Marquès-Bonet et al. (18), leurs génomes ne divergent que de 1%. Or, d'après le site web de Entrez Gene du NCBI (2006), la taille de ces génomes est d'environ 24 000 gènes par génome. Ce qui donne en moyenne 576 millions d'intervalles communs entre ces deux espèces.

#### 1.4 Différents types de scénarios

Lorsqu'on veut transformer une permutation  $P$  en une autre permutation  $Q$ , il existe une multitude de scénarios possibles. Le critère couramment utilisé pour choisir certains scénarios plutôt que d'autres est la longueur du scénario. En effet, on favorise les scénarios avec le moins de réarrangements possibles.

Cependant, il y a deux problèmes avec cette façon de procéder. Premièrement, les scénarios de longueur minimale sont souvent nombreux. Alors la question revient à savoir lequel, parmi la multitude de scénarios de longueur minimale, est le *vrai* scénario ?

Deuxièmement, les scénarios de longueur minimale ne sont pas toujours vraisemblables au point de vue biologique. En effet, il arrive qu'ils brisent les structures combinatoires communes alors qu'on est supposé les préserver. Dans ce cas, doit-on garder un scénario de longueur minimale qui brise les structures communes, ou le rejeter et chercher un autre scénario, peut-être en peu plus long, mais qui préserve ces structures ?

Suite à cette ambiguïté qui entoure le choix d'un scénario, on définit plusieurs types de scénarios qui seront présentés dans la suite de cette section.

### 1.4.1 Les scénarios parfaits

Un scénario qui préserve les structures combinatoires communes aux permutations à comparer ne sera pas nécessairement de longueur minimale, cependant il sera plus vraisemblable du point de vue biologique.

**Définition 5.** Soit  $P$  une permutation signée. Un scénario  $S$  de  $P$  est dit *scénario parfait* s'il ne brise aucun intervalle commun (Figeac et Varré, 2004).

Par exemple, si on a une permutation  $P = (1 \bar{3} \bar{2} 5 4 6)$ , on peut la transformer en identité en lui appliquant les inversions suivantes successivement  $\{2,3\}$ ,  $\{4,5\}$ ,  $\{4\}$ , et  $\{5\}$ . Dans la Figure 1.8, nous avons un scénario  $S$  de longueur 4 qui ne brise aucun intervalle commun de  $P$ . Un tel scénario est dit parfait.

inversions de $S$		permutation résultante
		$P = (1 \bar{3} \bar{2} 5 4 6)$
$\{2, 3\}$	$\implies$	$(1 2 3 5 4 6)$
$\{4, 5\}$	$\implies$	$(1 2 3 \bar{4} \bar{5} 6)$
$\{4\}$	$\implies$	$(1 2 3 4 \bar{5} 6)$
$\{5\}$	$\implies$	$(1 2 3 4 5 6)$

**FIG. 1.8** Un scénario qui trie  $P$  vers  $Id_n$  sans briser d'intervalle commun, c'est un scénario parfait.

Dans l'article de Figeac et Varré (11), les auteurs démontrent que, pour une permutation signée  $P$  donnée, il existe toujours un scénario parfait qui trie  $P$  vers  $Id_n$ .

### 1.4.2 Les scénarios parcimonieux

La *distance d'inversion* d'une permutation signée  $P$  est le nombre minimal d'inversions nécessaires pour transformer la permutation  $P$  en l'identité (Sankoff, 1992). Dans l'an-



nexe A, nous discutons de cette notion et nous donnons une méthode simple et facile pour calculer une borne inférieure de la distance d'inversion d'une permutation signée.

Par exemple, d'après cette méthode, la distance d'inversion de la permutation  $P = (1 \bar{3} \bar{2} 5 4 6)$  est supérieure ou égale à 3, ce qui veut dire que le scénario  $S$  de la Figure 1.8 n'est pas nécessairement le plus court scénario qui trie  $P$ .

En effet, si notre but était de transformer la permutation  $P = (1 \bar{3} \bar{2} 5 4 6)$  en l'identité sans se soucier de préserver les intervalles communs, nous aurions pu le faire dans un scénario  $S'$  de longueur 3, donc plus court que  $S$  mais, qui brise des intervalles communs. Le scénario  $S'$  est illustré dans la Figure 1.9, ci-dessous.

inversion de $S'$		permutation résultante		intervalles communs brisés
		$P = (1 \bar{3} \bar{2} 5 4 6)$		
$\{2, 3, 5\}$	$\Rightarrow$	$(1 \bar{5} 2 3 4 6)$	$\Rightarrow$	$\{1, 2, 3\}, \{4, 5\}$ et $\{4, 5, 6\}$
$\{2, 3, 4\}$	$\Rightarrow$	$(1 \bar{5} \bar{4} \bar{3} \bar{2} 6)$	$\Rightarrow$	
$\{2, 3, 4, 5\}$	$\Rightarrow$	$(1 2 3 4 5 6)$	$\Rightarrow$	

**FIG. 1.9** Un scénario *parcimonieux*  $S$  de  $P = (1 \bar{3} \bar{2} 5 4 6)$  vers l'identité. La longueur du scénario  $S'$  est égale à la distance d'inversion de  $P$ , ce qui implique que  $S'$  est un scénario *parcimonieux*. Cependant, il n'est pas parfait, car il brise des intervalles communs.

**Définition 6.** Soit  $P$  une permutation signée. Un scénario  $S$  de  $P$  est dit *scénario parcimonieux* s'il est de longueur minimale, ce qui veut dire que la longueur de  $S$  est égale à la distance d'inversion de  $P$ .

### 1.4.3 Les scénarios parfaits et parcimonieux

Lorsqu'un scénario *parcimonieux* ne brise aucun intervalle commun, on l'appelle alors un *scénario parfait et parcimonieux*. La Figure 1.10 nous donne un exemple tiré de l'article de Figeac et Varré (11) d'un tel scénario. Ce scénario trie une permutation  $P = (\bar{5} 1 \bar{9} \bar{8} \bar{7} 4 6 \bar{3} \bar{2})$  vers l'identité sans briser aucun intervalle commun de  $P$ . De

plus, selon cet article, la distance d'inversion de  $P$  est égale à la longueur de ce scénario.

$$\begin{aligned} r_1 : & ( \overline{5} \ 1 \ \overline{9} \ \overline{8} \ \overline{7} \ 4 \ 6 \ \overline{3} \ \overline{2} ) \\ r_2 : & ( \overline{5} \ 1 \ 2 \ 3 \ \overline{6} \ \overline{4} \ 7 \ 8 \ 9 ) \\ r_3 : & ( \overline{5} \ \underline{1 \ 2 \ 3 \ 4} \ 6 \ 7 \ 8 \ 9 ) \\ r_4 : & ( \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ 6 \ 7 \ 8 \ 9 ) \\ r_5 : & ( 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 ) \end{aligned}$$

**FIG. 1.10** Un scénario parfait et parcimonieux qui trie  $P$  vers l'identité. La permutation  $P$  possède les intervalles communs non-triviaux suivants :  $\{7, 8, 9\}$ ,  $\{7, 8\}$ ,  $\{8, 9\}$  et  $\{2, 3\}$ . Tous ces intervalles communs sont conservés durant tout le scénario. En plus, la distance d'inversion de  $P$  est 4 selon l'article de Figeac et Varré (11), donc ce scénario est parcimonieux.

Ces scénarios sont les plus intéressants, car ils satisfont aux deux critères de sélection, soit la longueur minimale et la conservation des intervalles communs. Sachant qu'un scénario parfait existe toujours, la difficulté de trouver un scénario parfait et parcimonieux repose sur son aspect parcimonieux.

#### 1.4.4 Les scénario parfaits les plus courts

Malheureusement, il arrive que tout scénario parcimonieux d'une permutation signée  $P$  brise au moins un intervalle commun de  $P$ . Dans ce cas, les deux critères ne peuvent pas être satisfaits simultanément. Il faut alors décider du critère à privilégier, soit la longueur minimale, soit la conservation des structures communes. Dans le présent travail, nous allons toujours privilégier la conservation des structures communes.

Ainsi, nous allons d'abord vérifier si, parmi les scénarios parcimonieux, il n'en existe pas au moins un qui soit parfait. S'il n'en existe pas, alors nous allons chercher le *scénario parfait le plus court* de tous les scénarios parfaits.

Notons que la longueur du scénario parfait le plus court sera plus grande ou égale à celle des scénarios parcimonieux.

Prenons l'exemple de la permutation  $P = (\bar{2} \bar{3} 1)$  dont la distance d'inversion est égale à 2. Cette permutation constitue un cas particulier car elle possède un seul scénario parcimonieux. En effet, normalement il y a plusieurs scénarios parcimonieux possibles pour une permutation donnée. Ce scénario unique de  $P$  est présenté dans la Figure 1.11.

$$\begin{aligned} r_1 : & ( \bar{2} \ \bar{3} \ \underline{1} ) \\ r_2 : & ( \bar{2} \ \underline{\bar{1}} \ 3 ) \\ r_3 : & ( 1 \ 2 \ 3 ) \end{aligned}$$

**FIG. 1.11** Le seul scénario parcimonieux de  $P$ . Cependant, il brise le seul intervalle commun non-trivial de  $P$ , soit  $\{2, 3\}$ , à la première inversion. Donc, comme il n'est pas parfait, nous allons le rejeter.

Sachant que notre priorité est la conservation des structures communes, nous allons rejeter le scénario de la Figure 1.11 car il brise des intervalles communs. Le scénario parfait le plus court de cette permutation est composé de 4 inversions et il est présenté dans la Figure 1.12.

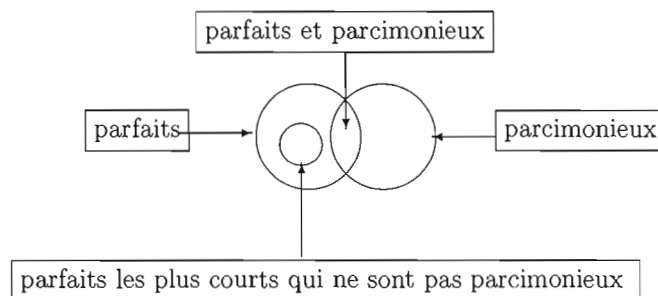
Notons que, dans ce scénario,  $P$  est triée vers l'identité renversée et non l'identité. En effet, comme c'est mentionné dans la Définition 1, un scénario de  $P$  trie la permutation  $P$  vers soit l'identité, soit l'identité renversée. Cet exemple illustre le cas où il est préférable de trier  $P$  vers l'identité renversée.

$$\begin{aligned} r_1 : & ( \bar{2} \ \bar{3} \ \underline{1} ) \\ r_2 : & ( \bar{2} \ \underline{\bar{3}} \ \bar{1} ) \\ r_3 : & ( \underline{\bar{3}} \ 2 \ \bar{1} ) \\ r_4 : & ( \bar{3} \ \underline{2} \ \bar{1} ) \\ r_5 : & ( \bar{3} \ \bar{2} \ \bar{1} ) \end{aligned}$$

**FIG. 1.12** Le scénario parfait le plus court qui trie la permutation  $P = (\bar{2} \bar{3} 1)$  vers l'identité renversée. Il est de longueur 4 alors que la distance d'inversion de  $P$  est de 2.

Dans ce travail, notre but principal est de trouver, pour une permutation signée donnée, un scénario qui ne brise aucun intervalle commun et qui soit le plus court possible. Dans certains cas, il arrive qu'on trouve un scénario parfait et parcimonieux. C'est la solution idéale, nos deux critères de sélection de scénario sont satisfaits. Cependant, dans d'autres cas, tous les scénarios parcimonieux sont non-parfaits.

Dans ces cas, il faut trouver le scénario parfait le plus court possible, car notre priorité est la conservation des structures combinatoires communes. La Figure 1.13 illustre la relation entre les différents types de scénarios.



**FIG. 1.13** La relation entre les différents types de scénarios d'une permutation signée.

Notons qu'un scénario parfait et parcimonieux est un cas particulier d'un scénario parfait le plus court de tous les scénarios parfaits. Dans la suite de ce travail, le terme "scénario parfait et parcimonieux" sera utilisé abusivement pour désigner le scénario parfait le plus court.

## CHAPITRE II

### DES INTERVALLES COMMUNS AUX INTERVALLES FORTS

Nous avons vu, dans le chapitre précédent, que les intervalles communs représentent, en bioinformatique, les structures conservées que nous voulons préserver dans des scénarios d'évolution. Nous avons vu aussi que le nombre d'intervalles communs d'une permutation sur  $n$  éléments est de  $O(n^2)$ , ce qui est très élevé, vu la taille des données en biologie. Dans ce chapitre, nous allons introduire la notion d'*intervalle fort*.

Les intervalles forts sont un sous-ensemble des intervalles communs qui ont l'avantage de représenter les intervalles biologiquement pertinents et d'être moins nombreux que les intervalles communs. De plus, ils ont la propriété de s'organiser sous forme d'un arbre. Cette représentation condensée permet de réduire l'espace.

Le fait d'être un intervalle commun ou un intervalle fort pour un intervalle donné  $I$  n'a aucun lien avec les signes des éléments de  $I$ . Ainsi, les résultats structuraux présentés dans ce chapitre sont autant valides pour les permutations signées que pour les permutations non signées. Pour des raisons de simplicité, nous allons omettre les signes des éléments dans ce chapitre et nous les réintégrerons dans le chapitre suivant.

#### 2.1 Intervalles forts

Nous introduisons d'abord, une relation fondamentale entre les intervalles d'une permutation.

**Définition 7.** Deux intervalles distincts  $I$  et  $J$  *commutent* si  $I \subset J$ , ou  $J \subset I$  ou encore

$I \cap J = \emptyset$ . Si les intervalles  $I$  et  $J$  ne commutent pas, on dit qu'ils se *chevauchent*. Aussi, une famille d'intervalles est dite *commutante* si chaque paire d'intervalles de la famille commute.

Parmi les intervalles communs d'une permutation donnée, certains se chevauchent et d'autres commutent. La Figure 2.1 nous montre les intervalles communs de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . Dans (a), nous y voyons tous les intervalles communs de  $P$ . Dans (b), nous avons les intervalles communs qui en chevauchent d'autres. Enfin, dans (c), nous avons les intervalles communs qui commutent avec tous les autres.

Pour alléger la figure, les singletons ne sont représentés ni parmi les intervalles communs ni parmi les intervalles communs et commutants.

$$\begin{array}{ll}
 P = ( \begin{array}{ccccccccc} 1 & 4 & 2 & 5 & 3 & 7 & 8 & 6 & 9 \end{array} ) & (a) \\
 \quad \quad \quad \begin{array}{ccccccccc} \hline \hline \hline \hline \hline \hline \hline \hline \hline \end{array} & \\
 P = ( \begin{array}{ccccccccc} 1 & 4 & 2 & 5 & 3 & 7 & 8 & 6 & 9 \end{array} ) & (b) \\
 \quad \quad \quad \begin{array}{ccccccccc} \hline \hline \hline \hline \hline \hline \hline \hline \hline \end{array} & \\
 P = ( \begin{array}{ccccccccc} 1 & 4 & 2 & 5 & 3 & 7 & 8 & 6 & 9 \end{array} ) & (c) \\
 \quad \quad \quad \begin{array}{ccccccccc} \hline \hline \hline \hline \hline \hline \hline \hline \hline \end{array} & 
 \end{array}$$

**FIG. 2.1** Les intervalles communs de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$  sont présentés dans (a). Dans (b), nous avons la représentation des intervalles communs de  $P$  qui en chevauchent d'autres. Dans (c), nous avons les intervalles communs de  $P$  qui commutent avec tous les autres.

**Définition 8.** Soit une permutation  $P$  sur  $n$  éléments, un *intervalle fort* de la permutation  $P$  est un intervalle commun de  $P$  qui commute avec tous les autres intervalles communs de  $P$ . Notons que, par définition, les singletons et l'ensemble  $\{1, 2, \dots, n\}$  sont des intervalles forts, ils sont dits *intervalles forts triviaux*.

Ainsi, nous avons un exemple d'intervalles forts dans la Figure 2.1 (c). En effet, chacun des intervalles communs de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$  présentés

dans (c) commute avec tous les intervalles communs de  $P$ , donc ce sont les intervalles forts de  $P$ , seuls les singletons ne sont pas représentés.

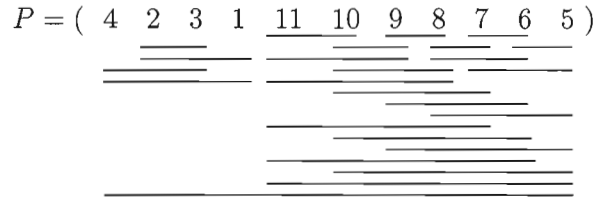
La famille des intervalles forts d'une permutation quelconque est commutante, car chaque paire d'intervalles forts commute. Ceci nous permet de les organiser sous forme d'un arbre d'inclusion qui sera présenté dans la section 2.2 .

Maintenant, nous allons discuter des avantages des intervalles forts par rapport aux intervalles communs en général.

### 2.1.1 Les intervalles forts sont moins nombreux que les intervalles communs

Un des avantages des intervalles forts, par rapport aux intervalles communs, est qu'ils sont moins nombreux. Ce qui implique qu'ils occupent moins d'espace tout en conservant toute l'information nécessaire. En effet, les intervalles forts sont les intervalles communs les plus significatifs.

Dans les Figures 2.2 et 2.3 qui suivent, nous avons un exemple qui montre à quel point nous sauvons de l'espace lorsque nous travaillons avec les intervalles forts au lieu des intervalles communs. En effet, la permutation  $P = (4\ 2\ 3\ 1\ 11\ 10\ 9\ 8\ 7\ 6\ 5)$  possède 24 intervalles communs non-triviaux, dont seulement trois sont forts.



**FIG. 2.2** La représentation des 25 intervalles communs de  $P = (4\ 2\ 3\ 1\ 11\ 10\ 9\ 8\ 7\ 6\ 5)$ . Seuls les singletons ne sont pas représentés.

Ceci est dû au fait que la plupart des intervalles communs sont liés à un tel point qu'il y en a qui sont moins intéressants que d'autres. Si on prend la permutation de la

Figure 2.2 par exemple, les éléments contenus dans l'intervalle (5..11) sont triés en ordre décroissant.

En effet, chaque suite d'éléments consécutifs dans cet intervalle constitue un intervalle commun de  $P$ . Ainsi, nous avons 20 intervalles communs non triviaux juste dans ce segment de la permutation. Or, la seule information nécessaire, pour cette section, est qu'elle est triée et l'intervalle fort (5..11), représenté dans la Figure 2.3, illustre complètement ce fait à lui seul, ainsi les 19 autres intervalles sont peu significatifs.

$$P = ( \quad 4 \quad \underline{\underline{2 \quad 3}} \quad 1 \quad \underline{\underline{11 \quad 10 \quad 9 \quad 8 \quad 7 \quad 6 \quad 5}} \quad )$$

**FIG. 2.3** Parmi les 24 intervalles communs non-triviaux de  $P = (4 \ 2 \ 3 \ 1 \ 11 \ 10 \ 9 \ 8 \ 7 \ 6 \ 5)$  représentés dans la Figure 2.2, seulement trois sont forts.

Nous montrerons, à la Section 2.2.2, la relation exacte entre le nombre d'intervalles communs et le nombre d'intervalles forts d'une permutation.

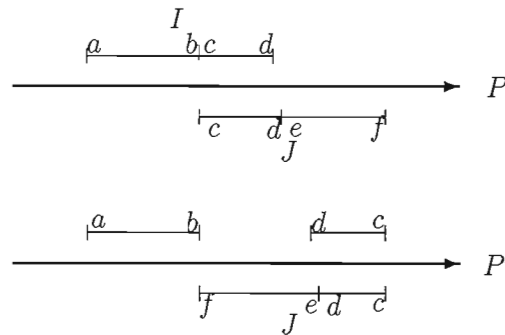
### 2.1.2 Inverser un intervalle fort ne brise jamais un intervalle commun d'une permutation

Si un intervalle  $I$  est un intervalle d'une permutation  $P$  et que  $J$  est un intervalle de  $P$  qui chevauche  $I$ , alors en inversant l'intervalle  $J$  de la permutation  $P$ , on aura une permutation résultante  $P'$  telle que  $I$  n'est pas un intervalle de  $P'$ . La Figure 2.4 illustre ce fait.

En effet, en observant la Figure 2.4, on remarque que les éléments de l'intervalle  $[f, \dots, e]$  dans  $P'$  s'insèrent entre les éléments de l'ancien intervalle  $I$ . Ce qui fait que  $I$  perd sa propriété d'intervalle dans la permutation  $P'$ .

Si, en plus, l'intervalle  $I$  était un intervalle commun de  $P$ , et que l'intervalle  $J$  appartient à un scénario  $S$  de la permutation  $P$ , alors nous dirons que le scénario  $S$  a brisé l'intervalle commun  $I$ . Ainsi l'ensemble d'intervalles communs de  $P$  n'est pas conservé durant le scénario  $S$ .





**FIG. 2.4** En inversant l'intervalle  $J$  qui chevauche l'intervalle  $I$  dans la permutation  $P$ , l'intervalle  $I$  n'est plus un intervalle dans la permutation résultante  $P'$ .

Nous avons vu, dans la Définition 5, qu'il ne faut pas briser un intervalle commun dans un scénario parfait. Aussi, par la Figure 2.4, nous constatons que, lorsque nous inversons un intervalle qui chevauche un intervalle commun d'une permutation, nous perdons cet intervalle commun.

Ainsi, désormais, nous allons éviter d'avoir une inversion qui chevauche un intervalle commun dans un scénario d'évolution. Or, par définition, les intervalles forts commutent avec tous les intervalles communs. Donc, les intervalles forts sont plus pertinents que les intervalles communs en général : en ajoutant un intervalle fort dans un scénario, on est certain de préserver tous les intervalles communs de la permutation.

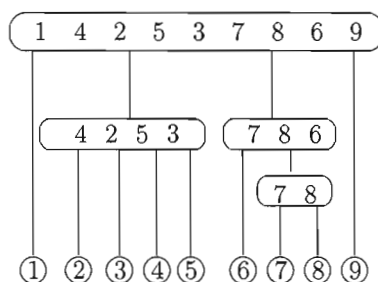
## 2.2 Arbre d'inclusion des intervalles forts

Dans cette section, nous présentons l'*arbre d'inclusion* des intervalles forts d'une permutation et ensuite nous verrons les propriétés qui découlent de cet arbre.

### 2.2.1 Présentation de l'arbre d'inclusion des intervalles forts

La famille des intervalles forts d'une permutation  $P$  est commutante. En effet, chaque paire d'intervalles forts  $P$  commute. Ainsi, l'ordre d'inclusion sur cette famille définit un *arbre d'inclusion* de  $n$  feuilles, noté  $T_f(P)$ , dont les noeuds sont les intervalles forts de la permutation  $P$ .

Prenons l'exemple de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$  de la Figure 2.1. Nous avons établi que les intervalles forts non-triviaux de  $P$  sont  $\{2, 3, 4, 5\}$ ,  $\{6, 7, 8\}$  et  $\{7, 8\}$ . Maintenant, la Figure 2.5 illustre l'arbre d'inclusion construits avec les intervalles forts de  $P$ . Notons que l'ordre des feuilles d'un arbre d'inclusion des intervalles forts est arbitraire.



**FIG. 2.5** L'arbre d'inclusion des intervalles forts  $T_f(P)$  de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . Les noeuds de l'arbre correspondent aux intervalles forts de  $P$ . Dans cet arbre, les feuilles sont triées selon la permutation identité. Cependant, ce choix est personnel.

Voici quelques caractéristiques de l'arbre d'inclusion d'intervalles forts d'une permutation quelconque  $P$ , noté  $T_f(P)$  :

1. chaque feuille de l'arbre  $T_f(P)$  correspond à un élément de la permutation  $P$ ,
2. la racine l'arbre  $T_f(P)$  correspond à la permutation elle-même,
3. le parent  $J$  d'un noeud  $I$  est le noeud qui correspond à l'intervalle fort le plus petit tel que  $I \subset J$ .

Nous avons déjà illustré, d'une façon informelle, que les intervalles forts sont moins nombreux que les intervalles communs. Maintenant, grâce aux propriétés de l'arbre d'inclusion des intervalles forts, nous allons montrer que, même si le nombre d'intervalles communs d'une permutation  $P$  sur  $n$  éléments est dans  $O(n^2)$ , le nombre d'intervalles forts de  $P$  est dans  $O(n)$ .

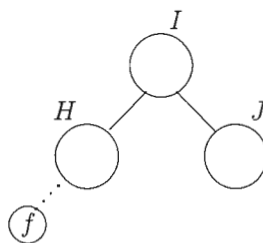
## 2.2.2 Les propriétés découlant de l'arbre d'inclusion

Dans cette section, nous allons montrer que les familles commutantes qui contiennent tous les singletons ont peu d'éléments. Ceci est une conséquence du fait que les noeuds de leur arbre d'inclusion ont toujours au moins deux enfants. Les résultats présentés dans cette section se trouvent dans les textes élémentaires sur les structures de données, Goodrich et Tamassia (13) en est un exemple.

**Proposition 1.** *Si une famille d'intervalles est commutante et contient tous les singletons alors chaque noeud interne de son arbre d'inclusion a au moins deux enfants.*

*Démonstration.* Soit un arbre d'inclusion  $T_f(P)$  d'une famille commutante et soit un noeud interne quelconque  $I$  et un autre noeud  $J$ , enfant de  $I$  (voir la Figure 2.6). Il existe au moins une feuille  $f$  qui appartient à  $I$  mais qui n'appartient pas à  $J$ ,  $f \in I \setminus J$  car, sinon,  $I$  est égal à  $J$  et  $J$  n'est pas l'enfant de  $I$ .

Donc il existe, nécessairement, un noeud  $H$  qui est le plus grand noeud inclus dans  $I$  qui contient la feuille  $f$  tel que  $H \neq I$ . Ainsi, le noeud  $I$  possède au moins deux enfants. □



**FIG. 2.6** Un arbre d'inclusion  $T_f(P)$  et ses noeuds  $I$  et  $J$  tels que  $J$  est l'enfant de  $I$ . Alors, il existe une feuille  $f$  telle que  $f \in I \setminus J$ . Soit un intervalle  $H$  tel que  $f \in H$  et  $H$  est le plus grand intervalle fort inclus dans  $I$  qui contient  $f$ . Donc,  $I$  possède au moins deux enfants.

**Lemme 1.** *Dans un arbre quelconque, le nombre d'arêtes de l'arbre est égal au nombre de noeuds de l'arbre moins un.*

*Démonstration.* Nous allons faire la démonstration par induction. Dans l'arbre qui a un seul noeud, il est évident que la proposition est vérifiée, car il n'y a pas d'arête. De la même manière, dans un arbre à deux noeuds, il y a une seule arête qui relie les deux noeuds.

Supposons maintenant que la proposition est vraie pour l'arbre  $T$  qui a  $m$  noeuds. Donc  $T$  possède  $(m - 1)$  arêtes. Soit  $T'$ , un arbre à  $(m + 1)$  noeuds. L'arbre  $T'$  possède au moins un noeud  $f$  qui est une feuille. Si, nous coupons la feuille  $f$  avec l'arête qui la relie à l'arbre, alors, nous aurons un arbre avec  $m$  noeuds. Or, par notre hypothèse inductive, un tel arbre possède  $(m - 1)$  arêtes. Ainsi, sachant que nous avons coupé une arête avec la feuille  $f$ , nous pouvons confirmer que l'arbre  $T'$  possédait  $m$  arêtes.  $\square$

**Définition 9.** Un arbre *binaire entier* est un arbre dont chaque noeud possède 0 ou 2 enfants.

**Lemme 2.** Dans un arbre binaire entier, le nombre d'arêtes est égal au double du nombre de noeuds internes.

*Démonstration.* Par définition, dans un arbre binaire entier, chaque noeud qui n'est pas une feuille possède 2 enfants. Donc, chaque noeud interne est relié à ses enfants par deux arêtes. Donc, si l'arbre possède  $n$  noeuds internes alors il doit nécessairement avoir  $2n$  arêtes.  $\square$

**Proposition 2.** Dans un arbre binaire entier, le nombre de noeuds internes est égal au nombre de feuilles moins un.

*Démonstration.* Soit  $B$ , le nombre d'arêtes,  $I$ , le nombre de noeuds internes et  $F$ , le nombre de feuilles. Par le Lemme 2, nous avons  $B = 2I$ , et par le Lemme 1,  $B = F + I - 1$ . Ce qui implique que  $I = F - 1$ .  $\square$

Ainsi, nous pouvons conclure que dans l'arbre d'inclusion des intervalles forts d'une permutation, le nombre de noeuds internes est égal au nombre de feuilles moins un au maximum.

Or nous savons déjà que, dans un arbre d'inclusion des intervalles forts, chaque élément de la permutation est une feuille. Ainsi, il y a autant de feuilles qu'il y a d'éléments dans la permutation. D'où la proposition suivante :

**Proposition 3.** *L'arbre d'inclusion d'intervalles forts d'une permutation  $P$  sur  $n$  éléments contient  $2n - 1$  noeuds au maximum.*

Nous venons ainsi de prouver que, si une permutation  $P$  sur  $n$  éléments possède  $O(n^2)$  intervalles communs, elle possède, par contre, seulement  $O(n)$  intervalles forts au pire cas, ce qui est une amélioration considérable, surtout si on pense à la taille des données en biologie.

## 2.3 L'arbre d'inclusion des intervalles forts avec les feuilles ordonnées

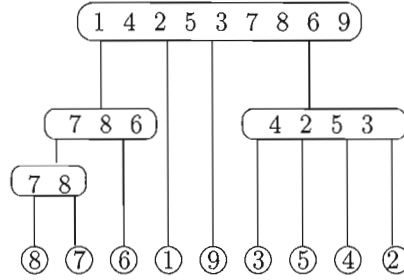
Dans la section précédente, nous avons discuté de l'arbre d'inclusion des intervalles forts. Maintenant, nous y ajoutons une contrainte : nous ordonnerons les feuilles de l'arbre selon une des permutations comparées. Ensuite, nous analyserons les conséquences de cet ordonnancement de feuilles sur l'arbre d'inclusion des intervalles forts au complet.

### 2.3.1 Ordonner les feuilles d'un arbre d'inclusion des intervalles forts

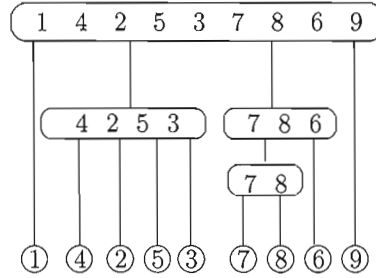
Revenons sur l'arbre d'inclusion de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$  présenté dans la Figure 2.6, nous avons déjà mentionné que le fait que ses feuilles soient ordonnées selon la permutation identité est un choix personnel. Ainsi, nous montrons une variante de cet arbre dans la Figure 2.7 avec un ordonnancement de feuilles différent.

Maintenant, nous ordonnons les feuilles de cet arbre selon l'une des deux permutations à comparer. Dans la Figure 2.8, les feuilles sont ordonnées selon la permutation  $P$  tandis que, dans la Figure 2.9, les feuilles sont ordonnées selon la permutation identité.

Remarquons que le fait d'ordonner des feuilles d'un arbre d'inclusion des intervalles forts d'une permutation n'affecte pas sa propriété d'arbre d'inclusion. Ainsi, les quatre arbres illustrés dans les Figures 2.5, 2.7, 2.8 et 2.9 sont tous des arbres d'inclusion des



**FIG. 2.7** L'arbre d'inclusion des intervalles forts  $T_f(P)$  de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . Les feuilles de cet arbre sont ordonnées d'une façon arbitraire.



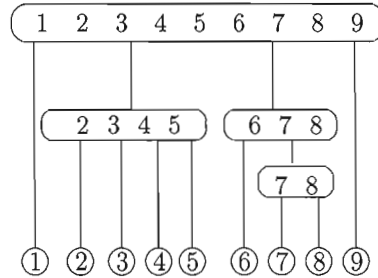
**FIG. 2.8** L'arbre d'inclusion des intervalles forts  $T_f(P)$  de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . Les feuilles de cet arbre sont ordonnées selon la permutation  $P$ .

intervalles forts valides de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ .

### 2.3.2 Les conséquences de l'ordonnancement des feuilles d'un arbre d'inclusion des intervalles forts

Soit  $I$  un intervalle commun d'une permutation  $P$  sur  $n$  éléments et  $x \in \{1, \dots, n\}$ . Si  $x \notin I$  alors, soit  $x$  est plus grand que tous les éléments de  $I$ , soit  $x$  est plus petit que tous les éléments de  $I$ . Dans les deux cas, nous utilisons les notations respectives  $x > I$  et  $x < I$ .

**Proposition 4.** *Soit deux intervalles communs disjoints  $I$  et  $J$  d'une permutation  $P$ , nous avons soit  $I < J$  qui veut dire que n'importe quel élément de  $I$  est plus petit que*



**FIG. 2.9** L'arbre d'inclusion des intervalles forts  $T_f(P)$  de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . Les feuilles de cet arbre sont ordonnées selon la permutation identité.

*n'importe quel élément de  $J$ , soit  $J < I$  qui veut dire que n'importe quel élément de  $J$  est plus petit que n'importe quel élément de  $I$ .*

Lorsque nous ordonnons les feuilles d'un arbre d'inclusion des intervalles forts d'une permutation, nous observons que cet ordre se propage d'une façon récursive dans tout l'arbre jusqu'aux enfants de la racine.

Prenons l'exemple de l'arbre de la Figure 2.9, les feuilles de cet arbre sont ordonnées selon la permutation identité donc du plus petit au plus grand élément de la permutation. Nous constatons que les noeuds internes d'un même parent sont aussi ordonnés selon l'ordre croissant des noeuds, tel que défini dans la Proposition 4.

Ainsi, par exemple, les enfants du noeud interne  $\{6, 7, 8\}$  apparaissent avec le noeud  $\{6\}$  en premier, suivi du noeud  $\{7, 8\}$ . De plus, les enfants de la racine apparaissent dans l'ordre suivant  $\{1\}$ ,  $\{2, 3, 4, 5\}$ ,  $\{6, 7, 8\}$  et  $\{9\}$ , soit l'ordre croissant des noeuds.

Dans cet exemple, les feuilles de l'arbre sont ordonnées selon la permutation identité. Cependant, nous faisons les mêmes observations lorsque nous ordonnons les feuilles selon la permutation  $P$  qui n'est pas l'identité. L'arbre de la Figure 2.8 constitue un exemple d'un tel cas.

Cette propagation récursive de l'ordre des feuilles dans l'arbre entier joue un rôle

important dans la résolution du problème de la recherche du scénario d'évolution entre deux espèces comme nous le verrons en détails dans la suite de ce travail.

Plus haut nous avons déjà mentionné que, pour des raisons de simplicité, l'une des permutations à comparer sera toujours la permutation identité et l'autre est une permutation quelconque sur le même ensemble d'éléments. Désormais, nous ordonnerons toujours les feuilles de l'arbre d'inclusion des intervalles forts selon la permutation qui n'est pas l'identité.

### 2.3.3 La permutation-quotient

Avant de continuer, nous avons besoin de définir la notion de *partition* d'un ensemble, car elle nous sera nécessaire pour comprendre la suite.

Une *partition* d'un ensemble  $\Sigma$  est un ensemble  $\Phi$  de sous-ensembles non vides de  $\Sigma$  deux à deux disjoints et qui forment un recouvrement de  $\Sigma$ . Autrement dit,  $\Phi$  est une partition de  $\Sigma$  si et seulement si les parties de  $\Phi$  sont non vides et tout élément  $x$  de  $\Sigma$  se trouve dans une et une seule partie de  $\Phi$ .

Prenons l'exemple d'un ensemble  $\Sigma = \{1, 2, 3, 4, 5\}$ . Les ensembles  $\{\{1, 4, 5\}, \{2, 3\}\}$  et  $\{\{1, 2\}, \{3\}, \{4, 5\}\}$  sont des partitions de  $\Sigma$ .

**Définition 10.** Une partition est dite *ordonnée* si on impose un ordre total sur ses éléments. Pour noter une partition ordonnée, nous utilisons des parenthèses au lieu des accolades.

Ainsi, par exemple, si la partition  $\{\{1, 2\}, \{3\}, \{4, 5\}\}$  de l'ensemble  $\{1, 2, 3, 4, 5\}$  est une partition ordonnée, elle sera notée  $(\{1, 2\}, \{3\}, \{4, 5\})$ .

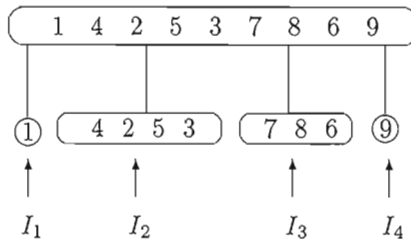
Remarquons que, dans un arbre d'inclusion des intervalles forts, l'ensemble des enfants d'un noeud interne constitue une partition sur l'ensemble correspondant à ce noeud. En effet, soit un noeud interne  $I$ , chaque élément de  $I$  est contenu dans un et un



seul enfant de  $I$  et l'union de tous les enfants de  $I$  donne l'ensemble  $I$ .

Aussi, nous pouvons ordonner les enfants du noeud  $I$  selon l'ordre croissant (ou décroissant) tel que défini dans la Proposition 4. Ainsi, supposons que le noeud  $I$  possède  $k$  noeuds enfants par exemple. Son plus petit enfant aura le rang 1, tandis que son plus grand aura le rang  $k$ .

Dans la Figure 2.10, nous avons un noeud  $I = (1, 4, 2, 5, 3, 7, 8, 6, 9)$ . Chaque enfant de  $I$  est étiqueté  $I_r$ , où  $r$  est son rang suite au classement des enfants de  $I$  par ordre croissant.



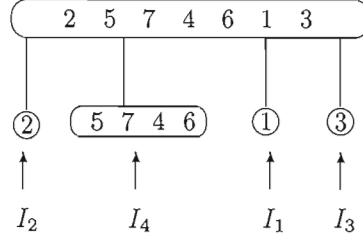
**FIG. 2.10** Nous accordons des rangs aux enfants du noeud  $(1, 4, 2, 5, 3, 7, 8, 6, 9)$  selon l'ordre croissant des noeuds. Par exemple, l'enfant  $\{1\}$  est le plus petit de tous, il est étiqueté  $I_1$ , le noeud  $(4, 2, 5, 3)$  est le deuxième plus petit, il est étiqueté  $I_2$ . Et ainsi de suite.

Dans la Figure 2.10, les enfants de la racine apparaissent dans l'ordre croissant. En effet, le plus petit noeud  $I_1$  apparaît en premier, suivi de  $I_2$ , puis de  $I_3$  et enfin  $I_4$ . Cependant, l'ordre d'apparition des enfants d'un noeud n'est pas toujours croissant. L'exemple de la Figure 2.11 montre un noeud dont les enfants ne sont pas triés.

**Définition 11.** Soit une permutation  $P$  sur  $n$  éléments et une partition  $\Phi = (I_1, \dots, I_k)$  de  $P$  formée par des intervalles forts de  $P$  et ordonnée par l'ordre croissant des intervalles forts de  $P$ . La *permutation-quotient* associée à  $\Phi$ , notée  $P|_{\Phi}$  est une permutation sur  $k$  éléments définie comme suit :

$$i \text{ précède } j \text{ dans } P|_{\Phi} \quad \text{si et seulement si} \quad I_i \text{ précède } I_j \text{ dans } P.$$

Prenons l'exemple de la permutation  $P = (\underline{1} \ 4 \ 2 \ 5 \ 3 \ 7 \ 8 \ 6 \ 9)$ . La partition suivante



**FIG. 2.11** Les enfants du noeud (2 5 7 4 6 1 3) apparaissent dans un ordre quelconque.

$\Phi = (\{1\}, \{2, 3, 4, 5\}, \{6\}, \{7, 8\}, \{9\})$  est formée par des intervalles forts de  $P$  et elle est ordonnée par l'ordre croissant des intervalles forts de  $P$ .

La permutation-quotient associée à  $\Phi$  est  $P|_{\Phi} = (1\ 2\ 4\ 3\ 5)$ . Ainsi, par exemple, sachant que l'intervalle  $I_3 = \{6\}$  et  $I_4 = \{7, 8\}$ , nous avons 4 qui précède 3 dans  $P|_{\Phi}$  puisque  $I_4$  précède  $I_3$  dans la permutation  $P$ .

Remarquons qu'une partition sur l'ensemble des éléments d'une permutation  $P$  formée d'intervalles forts de  $P$  n'est pas unique. En effet, outre la partition donnée dans l'exemple précédent, on peut aussi avoir la partition  $\Phi_1 = (\{1\}, \{2, 3, 4, 5\}, \{6, 7, 8\}, \{9\})$ . La permutation-quotient associée à cette partition est  $P|_{\Phi_1} = (1\ 2\ 3\ 4)$ .

Le lemme suivant relie les intervalles communs d'une permutation  $P$  à ceux d'une permutation-quotient définie à l'aide d'intervalles forts.

**Lemme 3.** *Posons  $\Phi = (I_1, \dots, I_k)$ , une partition ordonnée de la permutation  $P$  formée d'intervalles forts. L'intervalle  $J = \{j, \dots, h\}$  est un intervalle commun de la permutation-quotient  $P|_{\Phi}$  si et seulement si  $K = \bigcup_{j \leq i \leq h} I_i$  est un intervalle commun dans  $P$ .*

*Démonstration.* La démonstration de ce lemme va se faire en deux parties. Nous allons d'abord montrer que si  $J$  est un intervalle commun de  $P|_{\Phi}$  alors  $K$  est un intervalle commun de  $P$ . Ensuite, nous allons démontrer que si  $J$  n'est pas un intervalle commun de  $P|_{\Phi}$  alors  $K$ , non plus, n'est pas un intervalle commun de  $P$ .

Premièrement, supposons que  $J$  est un intervalle commun de  $P|_{\Phi}$ . Le fait que  $\Phi$  est une partition de  $P$  implique que chaque élément de  $P$  appartient à un et un seul intervalle de  $\Phi$ .

Soit  $x$  un élément de  $P$  tel que  $x \notin K$  et soit l'intervalle  $I_{\ell}$  de la partition qui contient  $x$ . Comme  $x \notin K$ , on a  $\ell \notin J$ .

Supposons qu'il existe deux éléments de  $P$ ,  $y, y' \in K$ , tels que  $y < x < y'$ . Étant donné que  $x$  n'appartient pas à  $K$  et qu'il est situé entre  $y$  et  $y'$  alors  $y$  et  $y'$  ne peuvent pas appartenir à un même intervalle de  $\Phi$ .

Alors, soit  $I_h$  et  $I_{h'}$ , les intervalles de  $\Phi$  qui contiennent respectivement  $y$  et  $y'$ ,  $I_h$  et  $I_{h'}$  sont distincts et  $h, h' \in J$ .

Il s'en suit que, dans la permutation-quotient  $P|_{\Phi}$ , on a  $h < \ell < h'$ , ce qui contredit le fait que  $J$  est un intervalle commun de  $P|_{\Phi}$ .

Deuxièmement, supposons que  $J$  n'est pas un intervalle commun de  $P|_{\Phi}$ . Alors, il existe les éléments  $\ell \notin J$  et  $h, h' \in J$  tels que  $h < \ell < h'$ . Ainsi,  $\forall y \in I_h$  et  $\forall y' \in I_{h'}$ ,  $y, y' \in K$  et il existe  $x \in I_{\ell}$  (i.e.,  $x \notin K$ ) tel que  $y < x < y'$ , ce qui fait que  $K$  n'est pas un intervalle commun dans  $P$ .  $\square$

**Définition 12.** Soit une permutation  $P$  et un intervalle fort  $I$  de  $P$ . L'intervalle  $I$  est dit *maximal* s'il n'existe aucun autre intervalle fort  $J$ , différent de l'intervalle  $\{1, \dots, n\}$ , tel que  $I \subset J$ .

Si on retourne à l'exemple de la permutation  $P = (1 \ 4 \ 2 \ 5 \ 3 \ 7 \ 8 \ 6 \ 9)$ , on a déjà établi que ses intervalles forts sont  $\{2,3,4,5\}, \{6,7,8\}, \{7,8\}$ , plus les intervalles forts triviaux. Parmi eux, les intervalles forts maximaux sont  $\{1\}, \{2,3,4,5\}, \{6,7,8\}, \{9\}$ . L'intervalle fort  $\{7,8\}$  n'est pas maximal, car il est inclus dans  $\{6,7,8\}$ . Pour la même raison, tous les singletons, sauf  $\{1\}$  et  $\{9\}$ , ne sont pas des intervalles forts maximaux.

Bref, pour une permutation  $P$  donnée, les intervalles forts maximaux de  $P$  sont

représentés par les enfants de la racine de son arbre d'inclusion.

**Lemme 4.** *Soit une permutation signée  $P$ , chaque intervalle commun de  $P$  est soit inclus dans un intervalle fort maximal ou c'est une union d'intervalles forts maximaux.*

*Démonstration.* Soit  $K$  un intervalle commun de  $P$ . Supposons que  $K$  n'est pas inclus dans un intervalle fort maximal. Comme  $K$  ne peut chevaucher un intervalle fort, par la définition même d'un intervalle fort, alors il ne peut pas, non plus, chevaucher un intervalle fort maximal. Ainsi,  $K$  est une union d'intervalles forts maximaux.  $\square$

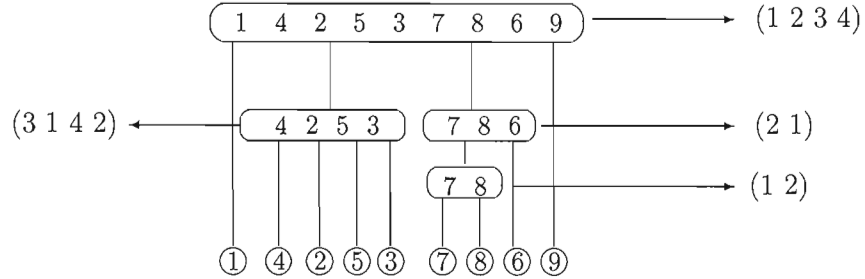
## 2.4 La classification des noeuds d'un arbre d'inclusion

Soit  $T_f(P)$ , l'arbre d'inclusion des intervalles forts entre la permutation  $P$  et la permutation identité sur le même ensemble que  $P$ . Nous avons mentionné, plus haut, que dans  $T_f(P)$  l'ensemble des enfants d'un noeud interne constitue une partition de l'ensemble correspondant à ce noeud.

De plus, nous avons décidé de toujours ordonner les feuilles selon la permutation différente de l'identité, soit  $P$ . Nous avons vu aussi que l'ordonnancement des feuilles de  $T_f(P)$  selon une des permutations à comparer se propage d'une façon récursive sur tous les noeuds de l'arbre. Ainsi, à chacun des noeuds internes d'un arbre d'inclusion correspond une permutation-quotient associée à la partition formée par ses enfants.

Dans la Figure 2.12, nous pouvons observer un arbre d'inclusion des intervalles forts de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$  dont les feuilles sont ordonnées selon leur ordre d'apparition dans la permutation  $P$ . À chaque noeud de l'arbre on associe sa permutation-quotient.

Nous allons maintenant présenter un théorème qui est fondamental pour la suite du document. Il énonce le fait que les permutations-quotients associées aux noeuds d'un arbre des intervalles forts sont soit des permutations triviales, c'est-à-dire des permutations de la forme  $(1\ 2\ \dots\ j-1\ j)$  ou  $(j\ j-1\ \dots\ 2\ 1)$ , soit des permutations qui ne contiennent aucun intervalle commun non-trivial.



**FIG. 2.12** L'arbre d'inclusion des intervalles forts de la permutation  $P = \{1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9\}$ . Chaque noeud interne est relié à une permutation-quotient associée à la partition formée par ses enfants.

Notons que les permutations qui ne contiennent aucun intervalle commun non-trivial sont "rares", au sens où elles sont difficiles à construire. En effet, une telle permutation contient d'abord au moins 4 éléments.

De plus, avec un ensemble de 4 éléments, soit  $\{1, 2, 3, 4\}$ , nous pouvons construire 24 permutations car, il y a  $n!$  permutations possibles sur un ensemble de  $n$  éléments. Sur ces 24 permutations, seulement 2 ne possèdent aucun intervalle commun non-trivial soit les permutations  $(3\ 1\ 4\ 2)$  et  $(2\ 4\ 1\ 3)$ .

Heureusement que de telles permutations soient rares car elles sont difficiles à trier. Nous reviendrons sur cet aspect dans le chapitre suivant.

**Théorème 1.** *Posons  $P$ , une permutation sur  $n$  éléments et  $\Phi = \{I_1, \dots, I_k\}$ , la partition de  $P$  formée d'intervalles forts maximaux de  $P$ . Alors,*

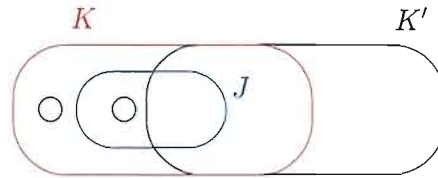
1. *soit chaque intervalle de la permutation-quotient  $P|_{\Phi}$  est un intervalle commun et dans ce cas, soit  $P|_{\Phi} = Id_k$  soit  $P|_{\Phi} = \overline{Id_k}$  ;*
2. *soit les seuls intervalles communs de  $P|_{\Phi}$  sont les intervalles communs triviaux.*

*Démonstration.* Dans les cas où  $k = 1, 2$  ou  $3$ , il est évident que le théorème est vérifié. Maintenant, nous allons démontrer que c'est aussi vrai lorsque  $k > 3$ .

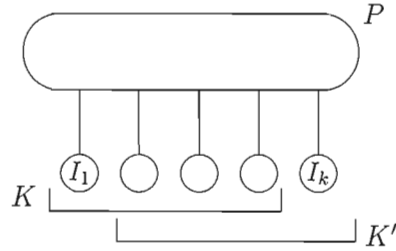
Supposons que  $P_{|\Phi}$  possède un intervalle commun non trivial. Soit  $I = \{i, \dots, j\}$ , l'intervalle commun non trivial le plus grand possible.

Soit  $K = \bigcup_{i \leq h \leq j} I_h$ . Alors  $K$  est un intervalle commun de  $P$ . Par définition de  $P_{|\Phi}$ ,  $K$  n'est pas fort. Il existe donc un autre intervalle commun  $K'$  qui chevauche  $K$ . Prenons un tel  $K'$  de taille maximale, avec  $I'$  l'intervalle correspondant dans  $P_{|\Phi}$ . Nous avons :

- a) l'ensemble  $K \cup K' = \{1, \dots, n\}$  sinon on aurait un intervalle commun contenant  $K$ , ce qui contredit la maximalité de  $K$ ,
- b) donc,  $I' \cup I = \{1, \dots, k\}$ ,
- c) ainsi, l'ensemble  $K' \setminus K$  ( et pour les mêmes raisons l'ensemble  $K \setminus K'$ ) contient un et un seul élément. Sinon,  $K' \setminus K$  est un intervalle non fort, ce qui implique qu'il existe un intervalle  $J$  qui chevauche  $K' \setminus K$  et  $J \cup K'$  est un intervalle commun plus grand que  $K'$ , ce qui contredit la maximalité de  $K'$ . La figure suivante illustre ce fait.



- d) Sachant que  $I \setminus I'$  et  $I' \setminus I$  sont des singletons et que  $I' \cap I$  est un intervalle commun de  $P_{|\Phi}$ , alors  $I \cap I'$  ne peut qu'être égal à  $\{2, \dots, k-1\}$ . C'est la seule façon de découper l'intervalle  $\{1, \dots, k\}$  en deux intervalles communs chevauchants avec un singleton à chaque bout. On remarque que  $K \cap K'$  est un intervalle commun aussi.
- e) On applique l'hypothèse d'induction à la permutation obtenue en retranchant  $I_1$  ou  $I_k$ , dépendamment de la position des grands éléments. La permutation résultante a un intervalle commun non trivial, notamment  $K \cap K'$  (si  $k \geq 3$ ).



□

Suite à ce théorème, nous pouvons classer les noeuds d'un arbre d'inclusion des intervalles forts dans trois catégories selon la permutation-quotient associée à la partition formée par ses enfants.

1. Soit les noeuds dont la permutation-quotient est une permutation identité, nous les appelons les noeuds *linéaires croissants*.
2. Soit les noeuds dont la permutation-quotient est une permutation identité inversée, nous les appelons les noeuds *linéaires décroissants*.
3. Soit les noeuds dont la permutation-quotient est une permutation dont les seuls intervalles communs sont les triviaux, nous les appelons les noeuds *premiers*.

Ainsi, par exemple, l'arbre de la Figure 2.12 comprend deux noeuds linéaires croissants à savoir la racine et le noeud  $\{7, 8\}$ , un noeud linéaire décroissant  $\{6, 7, 8\}$  et enfin un noeud premier  $\{2, 3, 4, 5\}$ .

## 2.5 Les arbres-PQ

Dans la théorie des graphes, il existe une structure de données appelée *arbre-PQ* qui possède les mêmes caractéristiques qu'un arbre d'inclusion des intervalles forts d'une permutation dont les feuilles sont ordonnées. Cette structure possède, en plus, d'autres propriétés qui sont intéressantes pour notre problème.

Dans cette section, nous allons présenter les arbres-PQ et les avantages qu'ils

apportent à la résolution du problème de calcul d'un scénario d'évolution qui ne brise pas les structures combinatoires communes.

### 2.5.1 Introduction aux arbres-PQ

Les arbres-PQ ont été introduits par Booth et Lueker (1976) pour représenter une classe de permutations sur un ensemble d'éléments. Ce sont des arbres enracinés qui possèdent trois sortes de noeuds : les *feuilles* qui sont des noeuds externes, puis les noeuds internes de type  $Q$ , modélisés par un rectangle et enfin, les noeuds internes de type  $P$  modélisés par un ovale. Voici quelques propriétés d'un arbre-PQ :

1. la lecture des feuilles d'un arbre-PQ de gauche à droite correspond à une des permutations de la classe,
2. les enfants d'un noeud  $P$  n'ont pas d'ordre particulier,
3. les enfants d'un noeud  $Q$  apparaissent dans un certain ordre qui doit être préservé.

Les arbres-PQ sont utilisés pour résoudre différents problèmes tels que déterminer si une matrice binaire possède la propriété des 1-consécutifs, tester si un graphe est un graphe d'intervalles ou vérifier sa planarité (Booth et Lueker, 1976).

En génomique, en plus de la détection de groupes de gènes fonctionnels (Landau et al, 2005), les arbres-PQ servent, par exemple, à établir l'ordre des longs segments d'ADN obtenus lors du séquençage d'un génome (Ito et al., 2006).

Les arbres-PQ peuvent servir aussi à représenter les intervalles forts entre les permutations à comparer dans le problème de recherche du scénario d'évolution. C'est cette dernière fonctionnalité qui nous intéresse dans ce travail.

### 2.5.2 Quelques définitions de base pour comprendre les arbres PQ

Dans cette section, nous présenterons des définitions et des propriétés concernant les arbres-PQ qui nous permettront d'étudier, d'une façon globale, cette structure de don-



nées.

Soit  $T$  un arbre-PQ sur un ensemble  $\Sigma$ . La *frontière* de  $T$  est la permutation sur l'ensemble  $\Sigma$  obtenue en parcourant les feuilles de l'arbre  $T$  de gauche à droite.

Il existe deux types d'opérations sur un arbre-PQ, qu'on appelle les *transformations équivalentes*, qui sont les suivantes :

1. réarranger les enfants d'un noeud de type  $P$  dans un ordre arbitraire car les enfants d'un noeud de type  $P$  ne sont pas ordonnés ;
2. contrairement au noeud de type  $P$ , les enfants d'un noeud de type  $Q$  peuvent être réarrangés seulement par un renversement du noeud au complet. Ceci implique que, après le réarrangement, les enfants aux extrémités le seront toujours et qu'un enfant du milieu aura toujours les mêmes voisins immédiats.

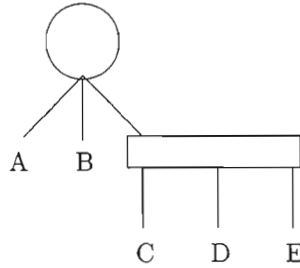
La classe de permutations représentée par un arbre-PQ  $T$  est composée de toutes les frontières pouvant être obtenues par des transformations équivalentes sur l'arbre  $T$ . L'ensemble de ces frontières est appelé l'ensemble des permutations *compatibles* avec l'arbre-PQ .

Dans la Figure 2.13, nous avons un exemple d'un arbre-PQ composé par un noeud de type  $P$  qui est la racine de l'arbre et un noeud de type  $Q$  illustré par un rectangle. L'arbre possède cinq feuilles. Dans le bas de la figure, nous avons l'ensemble des permutations compatibles avec cet arbre.

### 2.5.3 Des arbres d'inclusion aux arbres PQ

Maintenant, nous allons établir un parallèle entre l'arbre d'inclusion avec les feuilles ordonnées et l'arbre-PQ. Nous avons vu que les deux structures possèdent trois types de noeuds. Dans les deux arbres, les feuilles correspondent aux éléments de l'ensemble sur lequel la classe de permutations est définie.

Concernant les noeuds internes, nous observons des similarités entre un noeud



Voici l'ensemble des permutations compatibles de l'arbre :

$\{ (A B C D E), (A C D E B), (B A C D E), (B C D E A), (C D E A B), (C D E B A), (A B E D C), (A E D C B), (B A E D C), (B E D C A), (E D C A B), (E D C B A) \}$

**FIG. 2.13** Cet arbre-PQ représente une classe de permutations. Cette classe est un sous-ensemble de l'ensemble de toutes les permutations possibles sur l'ensemble  $\{A,B,C,D,E\}$ .

linéaire d'un arbre d'inclusion et un noeud de type  $Q$  d'un arbre-PQ. En effet, dans ces deux types de noeud, les enfants sont triés et cet ordre doit être préservé.

Aussi, autant dans un noeud premier d'un arbre d'inclusion que dans un noeud de type  $P$  de l'arbre PQ, les enfants ne sont pas ordonnés, ce qui laisse toute la latitude pour les opérations qu'on peut effectuer sur ce type de noeud.

De plus, comme dans un arbre d'inclusion des intervalles forts avec feuilles ordonnées selon l'une des permutations comparées, les feuilles d'un arbre-PQ sont triées selon leur ordre d'apparition dans l'une des permutations de la classe représentée par l'arbre.

Par ailleurs, les arbres-PQ sont intéressants pour notre problème grâce à certaines opérations définies sur cette structure à savoir les transformations équivalentes. En effet, ces transformations nous permettent de trier une permutation par des réarrangements sans briser les intervalles communs, étant donné que la seule opération permise sur un noeud de type  $Q$  est le renversement du noeud au complet.

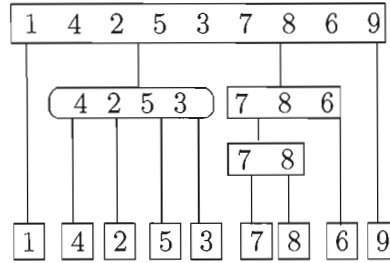
Ainsi, au lieu de concevoir de nouvelles opérations pour l'arbre d'inclusion des intervalles forts avec les feuilles ordonnées, nous pouvons utiliser la structure de données

d'arbre-PQ déjà existante et qui a déjà fait ses preuves.

Dans la suite de ce document, nous travaillerons avec les arbres-PQ des intervalles forts au lieu des arbres d'inclusion des intervalles forts et nous les appellerons simplement les arbres des intervalles forts. Les noeuds linéaires de l'arbre d'inclusion seront représentés par les noeuds de type  $Q$ , donc illustrés par des rectangles, et les noeuds premiers par des noeuds de type  $P$ , donc illustrés par des ovals.

Notons que, lorsque nous représentons les intervalles forts entre deux permutations  $P$  et  $Q$  par un arbre-PQ des intervalles forts  $T$ , et que nous voulons trier l'une des permutations vers l'autre, disons  $P$  vers  $Q$ , alors les permutations  $P$ ,  $Q$  et toutes les permutations du scénario de  $P$  vers  $Q$  appartiennent, obligatoirement, à l'ensemble des permutations compatibles de l'arbre  $T$ .

Dans la Figure 2.14, on a un exemple d'arbre des intervalles forts de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ .



**FIG. 2.14** L'arbre des intervalles forts  $T_f(P)$  de la permutation  $P = (1\ 4\ 2\ 5\ 3\ 7\ 8\ 6\ 9)$ . Il comprend trois noeuds linéaires, dont la racine, et un noeud premier.

La proposition suivante nous aide à identifier tous les intervalles communs d'une permutation signée  $P$  à partir de l'arbre des intervalles forts  $T_f(P)$  de  $P$ .

**Proposition 5.** *Un intervalle  $I$  d'une permutation signée  $P$  est un intervalle commun si et seulement si c'est soit un noeud de l'arbre des intervalles forts  $T_f(P)$  ou l'union des enfants consécutifs d'un noeud linéaire de  $T_f(P)$ .*

*Démonstration.* Supposons que  $I$  est un intervalle commun de  $P$ . Si  $I$  est un noeud de

l'arbre des intervalles forts, alors il est évident que  $I$  est un intervalle commun, car tout intervalle fort est un intervalle commun.

Sinon,  $I$  est un intervalle commun qui n'est pas fort. Par définition d'un intervalle fort, il existe un plus petit intervalle fort  $J$  qui contient  $I$ . L'intervalle  $I$  commute avec tous les enfants de  $J$ , car les enfants de  $J$  sont des intervalles forts. Ce qui veut dire que,  $\forall H$  enfant de  $J$ , on a :

1. soit  $I \subset H$ ,
2. soit  $H \subset I$ ,
3. soit  $I \cap H = \emptyset$ .

Cependant, il y a un problème avec le premier cas, car si  $I \subset H$ , le fait que  $J$  est le plus petit intervalle fort qui contient  $I$  est contredit, car  $H$  l'est. Ainsi, aucun enfant de  $J$  ne satisfait cette condition.

Pour ce qui est du deuxième cas, il faut qu'il y ait des enfants de  $J$  qui satisfont cette condition, car sinon la condition de départ  $I \subset J$  ne sera pas vérifiée.

Quant au dernier cas, il implique que certains enfants de  $J$  n'appartiennent pas à  $I$ . Ce qui prouve que  $I$  est l'union d'un sous-ensemble d'enfants de  $J$ . Ces enfants doivent être consécutifs parce que  $I$  est un intervalle de  $P$ .

De plus, le Théorème 1 implique que  $J$  doit être linéaire. En effet, si  $J$  est premier, le deuxième point du Théorème 1 implique que n'importe quel sous-ensemble d'enfants de  $J$ , sauf les singletons, n'est pas un intervalle commun de  $P$ .  $\square$

Pour illustrer la Proposition 5, prenons l'arbre représenté dans la Figure 2.14. Si nous observons les enfants des noeuds linéaires, nous constatons que l'union des enfants consécutifs est un intervalle commun. Prenons la racine, par exemple, c'est un noeud linéaire. En faisant l'union de deux enfants les plus à gauche, on a  $\{1\} \cup \{2, 3, 4, 5\}$  ce qui donne  $\{1, 2, 3, 4, 5\}$  qui est un intervalle commun de  $P$ . Ceci est vrai pour n'importe

quelle union d'enfants voisins de ce noeud.

Cependant, si on analyse le seul noeud premier qu'on a, on constate que l'union des enfants consécutifs ne donne jamais un intervalle commun sauf si on prend tous les enfants pour avoir le noeud lui-même.

En effet, si on prend l'union des deux enfants les plus à gauche par exemple, on a  $\{4\} \cup \{2\}$  qui donne  $\{2, 4\}$  qui n'est pas un intervalle commun. En prenant les trois enfants les plus à gauche on a  $\{4\} \cup \{2\} \cup \{5\}$  qui donne  $\{2, 4, 5\}$  qui n'est pas un intervalle commun. En prenant les enfants les plus à droite, on a  $\{5\} \cup \{3\}$  qui donne  $\{3, 5\}$  qui n'est pas un intervalle commun. En prenant les trois enfants les plus à droite on a  $\{3\} \cup \{5\} \cup \{2\}$  qui donne  $\{2, 3, 5\}$  qui n'est pas un intervalle commun.

Par contre, si on fait l'union de tous les enfants du noeud, alors on a le noeud au complet  $\{2, 3, 4, 5\}$ , un intervalle commun car c'est un intervalle fort. Cette distinction de noeuds internes rend le problème de calcul de scénario d'évolution plus facile comme nous le verrons dans le chapitre suivant.

Cette représentation des intervalles forts avec les arbres-PQ nous sert de base pour générer tous les intervalles communs de la permutation  $P$  au besoin, sans avoir à stocker les  $O(n^2)$  intervalles communs au pire cas.

En effet, nous savons que chaque intervalle fort est un intervalle commun, ce qui implique que tout noeud de l'arbre  $T_f(P)$  est un intervalle commun. En plus, nous venons de voir que, d'après les propriétés de cet arbre, toute union d'enfants consécutifs d'un noeud linéaire est un intervalle commun de  $P$  aussi.

## CHAPITRE III

### IDENTIFIER UN SCÉNARIO PARFAIT ET PARCIMONIEUX

Nous avons discuté dans le chapitre précédent de l'arbre des intervalles forts. Dans ce chapitre, nous montrons que, même si le problème général de calcul du scénario parfait et parcimonieux est difficile (Figeac et Varré, 2004), il peut se résoudre efficacement pour certaines classes de permutations définies en terme de la structure de leur arbre des intervalles forts.

#### 3.1 Les arbres des intervalles forts

Dans cette section, nous verrons la façon d'accorder des signes aux noeuds d'un arbre des intervalles forts. Ensuite, nous parlerons de différents types d'arbres des intervalles forts signés. Afin, nous présenterons les données biologiques réelles qui nous serviront d'exemples dans nos explications.

##### 3.1.1 Accorder les signes aux noeuds de l'arbre

Dans le chapitre précédent, nous avons décidé d'omettre les signes des éléments des permutations à comparer, parce qu'ils ne jouaient aucun rôle dans le fait qu'un intervalle soit fort ou pas. Dans la présente section, nous reintégrons les signes aux éléments des permutations car ils sont nécessaires pour la suite de ce chapitre. De plus, nous étendons la notion de signe à des intervalles complets.

**Définition 13.** Un *arbre d'intervalles forts signé* d'une permutation signée est un arbre d'intervalles forts dont chaque noeud possède un signe. Le signe d'un noeud peut être *positif* (+), *négatif* (−) ou *indéfini*.

Rappelons que le but de ce travail est de trouver un scénario qui trie une permutation signée  $P$  sur  $n$  éléments vers la permutation identité  $Id_n$  ou vers la permutation identité inversée  $\overline{Id_n}$ .

Pour y arriver, nous utilisons l'arbre des intervalles forts de la permutation  $P$ . Dans cet arbre, la permutation qui est représentée par la racine est partitionnée dans des sous-permutations, les enfants de la racine, qui, à leur tour, sont partitionnés dans d'autres sous-permutations, et ainsi de suite jusqu'aux feuilles.

Ainsi, le défi devient maintenant de trier ces sous-permutations représentées par les noeuds de l'arbre au lieu de la permutation entière, ce qui rend le travail plus simple. Aussi, si tous les enfants d'un noeud sont triés, alors trier le noeud devient trivial.

Le fait d'accorder un signe à un noeud donné de l'arbre des intervalles forts de la permutation nous aide à savoir dans quel ordre ce noeud sera trié. En effet, si nous accordons le signe + à un noeud  $I$  de l'arbre, cela signifie que nous allons trier  $I$  dans l'ordre croissant, tandis que, si nous lui accordons un signe −, nous le trierons dans l'ordre décroissant. Nous accorderons un signe indéfini à un noeud pour signifier que nous ne savons pas encore dans quel ordre nous devrions le trier.

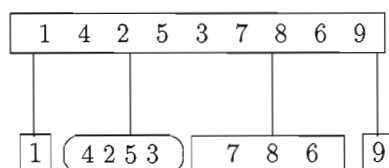
Maintenant, nous allons discuter du processus qui affecte des signes aux noeuds d'un arbre des intervalles forts d'une permutation signée quelconque. Le type du noeud auquel nous voulons donner un signe joue un rôle important. D'abord, nous verrons la manière d'accorder un signe à un noeud linéaire, puis à un noeud premier et enfin à une feuille.

D'après le Théorème 1, nous savons déjà qu'il y a deux types de noeuds linéaires soit les noeuds linéaires croissants dont la permutation-quotient est l'identité et les noeuds linéaires décroissants dont la permutation-quotient est l'identité inversée. Ainsi,

naturellement, nous affecterons le signe positif aux noeuds linéaires croissants et le signe négatif aux noeuds linéaires décroissants.

En ce qui concerne les noeuds premiers, c'est un peu plus complexe car, d'après la définition d'un noeud premier, il n'existe aucun ordre naturel sur ses enfants. Ainsi, pour savoir dans quel ordre nous allons le trier, il y a deux situations différentes, à savoir si son parent est un noeud linéaire ou non.

En effet, si le parent est linéaire, en observant les frères du noeud premier auquel nous voulons accorder un signe, intuitivement, nous saurons dans quel ordre nous devons le trier.



**FIG. 3.1** Intuitivement, nous savons que le noeud premier  $(4, 2, 5, 3)$  devrait être trié dans l'ordre croissant en observant ses frères car son parent est linéaire.

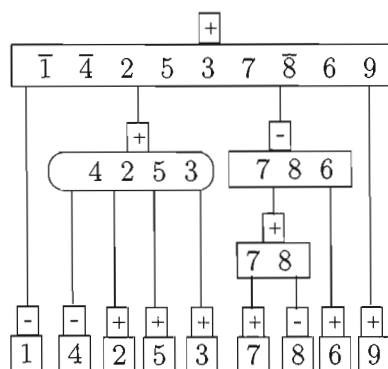
Prenons l'exemple du noeud premier  $(4, 2, 5, 3)$  dans l'arbre de la Figure 3.1. Son parent est linéaire. En observant ses frères, nous constatons que le seul frère situé à sa gauche,  $(1)$ , est plus petit que lui puis, que les deux frères situés à sa droite,  $(7, 8, 6)$  et  $(9)$ , sont plus grands que lui. Ainsi, nous en concluons qu'il serait plus naturel de trier le noeud  $(4, 2, 5, 3)$  dans l'ordre croissant. Ainsi, un noeud premier de parent linéaire va hériter le signe de son parent.

Par contre, si le parent du noeud premier n'est pas linéaire, c'est le cas d'un noeud premier qui est la racine ou d'un noeud premier dont le parent est aussi un noeud premier, alors nous ne pouvons pas savoir dans quel ordre trier un tel noeud. Par conséquent, le signe d'un noeud premier dont le parent n'est pas linéaire sera indéfini.

Enfin, nous accordons à une feuille le signe de l'élément de la permutation qui correspond à cette feuille.



Ainsi, après la construction de l'arbre des intervalles forts  $T_f(P)$  de la permutation, nous appliquons l'Algorithme 1 présenté ci-bas. Cet algorithme fait un parcours de l'arbre  $T_f(P)$  en profondeur en ordre préfixe, en affectant un signe à chaque noeud de l'arbre de la manière décrite ci-haut. À la fin du parcours, chaque noeud possède un signe et l'opération se fait en temps linéaire. La Figure 3.2 est un exemple d'arbre d'intervalles forts signé.



**FIG. 3.2** L'arbre signé des intervalles forts  $T_f(P)$  de  $P = (\bar{1} \bar{4} 2 5 3 7 \bar{8} 6 9)$ . La racine est linéaire croissant donc son signe est +. Le noeud linéaire  $(7, 8, 6)$  est de signe + car il est décroissant, le noeud linéaire  $(7, 8)$  est croissant donc de signe + et le noeud premier  $(4, 2, 5, 3)$  est de signe + car son parent est +. Les feuilles ont les signes des éléments correspondants dans  $P$ .

Pour conclure cette section, rappelons que Bergeron et al. (4) décrivent un algorithme qui construit les arbres des intervalles forts non signés en temps linéaire. Cet algorithme est implanté dans une application qui se trouve sur le site du Laboratoire de Génomique Comparée à l'adresse suivante <http://cgl.bioinfo.uqam.ca/>. Elle prend les permutations à comparer en entrée et elle construit l'arbre des intervalles forts de ces permutations.

Dans le présent travail, pour calculer les scénarios d'évolution, nous partons d'un arbre construit par cette application. Ensuite, nous affectons les signes aux noeuds de cet arbre de la manière décrite ci-haut, ce qui donne un arbre des intervalles forts signé.

**Algorithme 1** : L'algorithme qui affecte un signe au noeud  $I$  de l'arbre des intervalles forts passé en paramètre.

**Entrée** : L'arbre des intervalles forts  $T_f(P)$  non-signé d'une permutation  $P$ .

**Sortie** : L'arbre  $T_f(P)$  où chaque noeud est muni d'un signe  $+$ ,  $-$  ou indéfini.

```

1 : Visiter chaque noeud  $I$  de l'arbre  $T_f(P)$  par un parcours préfixe ;
2 :     si  $I$  est une feuille alors
3 :         le signe de  $I$  est le signe de l'élément qui lui correspond dans  $P$  ;
4 :     sinon si le noeud  $I$  est linéaire alors
5 :         si le noeud  $I$  est croissant alors
6 :             le signe de  $I$  est  $+$  ;
7 :         sinon
8 :             le signe de  $I$  est  $-$  ;
9 :         fin
10 :    sinon
11 :        si le parent de  $I$  est linéaire alors
12 :             $I$  hérite du signe de son parent ;
13 :        sinon
14 :            le signe de  $I$  est indéfini ;
15 :        fin
16 :    fin

```

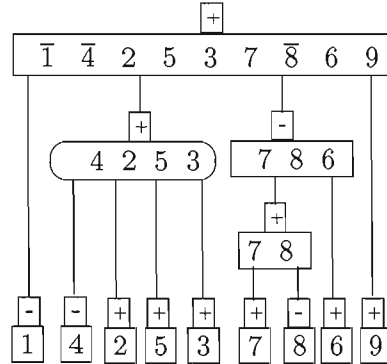
### 3.1.2 Différents types d'arbres

Nous pouvons classer les arbres des intervalles forts d'une permutation dans deux catégories différentes selon leur structure. Dans cette section, nous discuterons de cette classification, puis nous donnerons les exemples d'arbres pour chacune de ces classes.

**Définition 14.** Un arbre des intervalles forts est dit *non-ambigu* si tous ses noeuds premiers possèdent un parent linéaire, sinon il est dit *ambigu*. Si l'arbre ne possède aucun noeud premier alors il est *défini*. Notons qu'un arbre défini est un cas particulier

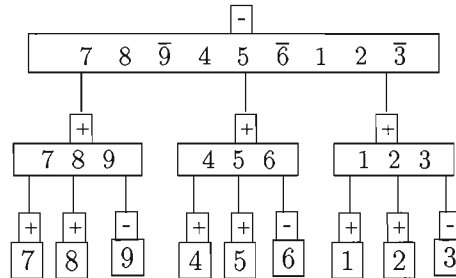
d'un arbre non-ambigu.

L'arbre de la Figure 3.3 est un exemple d'un arbre des intervalles forts non-ambigu. En effet, cet arbre possède un seul noeud premier  $(4, 2, 5, 3)$  dont le parent est linéaire.



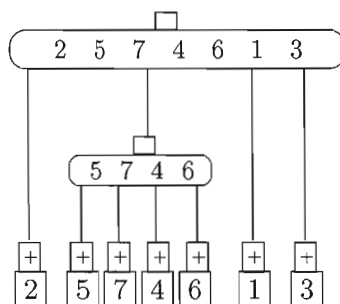
**FIG. 3.3** L'arbre des intervalles forts de la permutation  $P = (\bar{1} \bar{4} 2 5 3 7 \bar{8} 6 9)$ . Cet arbre est non-ambigu parce que son seul noeud premier  $(4, 2, 5, 3)$  possède un parent linéaire.

L'arbre des intervalles forts de la Figure 3.4 est défini car il ne possède aucun noeud premier. Tous ses noeuds internes sont linéaires.



**FIG. 3.4** L'arbre des intervalles forts de la permutation  $P = (7 8 \bar{9} 4 5 \bar{6} 1 2 \bar{3})$ . Il ne possède aucun noeud premier. C'est donc un arbre défini.

L'arbre de la Figure 3.5 représente un arbre des intervalles forts ambigu. Cet arbre possède deux noeuds premiers dont les parents ne sont pas des noeuds linéaires. En effet, la racine est un noeud premier tandis que le parent de l'autre noeud premier,  $(5, 7, 4, 6)$ , est la racine.



**FIG. 3.5** L'arbre des intervalles forts de la permutation  $P = (2\ 5\ 7\ 4\ 6\ 1\ 3)$ . Il contient deux noeuds premiers soit la racine et le noeud  $(5, 7, 4, 6)$  et les parents de ces deux noeuds ne sont pas linéaires. Ainsi, cet arbre est ambigu.

Dans un tel arbre, comme il n'y a pas d'ordre naturel sur la racine, qui est un noeud premier, il est difficile de décider si le noeud  $(5, 7, 4, 6)$ , par exemple, devrait être trié dans un ordre croissant ou décroissant. Aucune information ne nous aide à résoudre ce cas. Il y a une ambiguïté d'où le nom d'arbre ambigu.

Le type d'un arbre des intervalles forts joue un rôle important dans le calcul d'un scénario parfait et parcimonieux. Nous en discuterons dans la section 3.2.

### 3.1.3 La présentation des données

Dans l'article de Gibbs et al. (12), les auteurs présentent une étude comparative des séquences d'ADN qui composent les chromosomes X de l'homme, de la souris et du rat. Les résultats de cette étude ont suscité un intérêt particulier dans le problème de la recherche de scénarios d'évolution. En effet, ces trois chromosomes sont composés par 16 fragments qui sont communs dans les trois espèces à l'ordre près. Ces fragments s'appellent des *blocs de synténie*.

Ainsi, les chromosomes X de ces trois espèces peuvent être représentés par trois permutations distinctes sur l'ensemble  $\{1, 2, \dots, 16\}$ . Chaque bloc est représenté par un nombre entre 1 et 16.

$$\begin{aligned}
\text{ChrX}_H &= (\boxed{1} \quad \boxed{2} \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad 13 \quad 14 \quad 15 \quad 16) \\
\text{ChrX}_S &= (\bar{6} \quad \bar{5} \quad 4 \quad 13 \quad 14 \quad \bar{15} \quad 16 \quad \boxed{1} \quad \bar{3} \quad 9 \quad \bar{10} \quad 11 \quad 12 \quad \bar{7} \quad 8 \quad \boxed{\bar{2}}) \\
\text{ChrX}_R &= (\bar{13} \quad \bar{4} \quad 5 \quad \bar{6} \quad \bar{12} \quad \bar{8} \quad \bar{7} \quad \boxed{2} \quad \boxed{1} \quad \bar{3} \quad 9 \quad 10 \quad 11 \quad 14 \quad \bar{15} \quad 16)
\end{aligned}$$

**FIG. 3.6** Les permutations sur les 16 blocs de synténies des chromosomes X de la souris, du rat et de l'homme (qui sert de repère). Les blocs 1 et 2 sont encadrés chez les trois espèces.

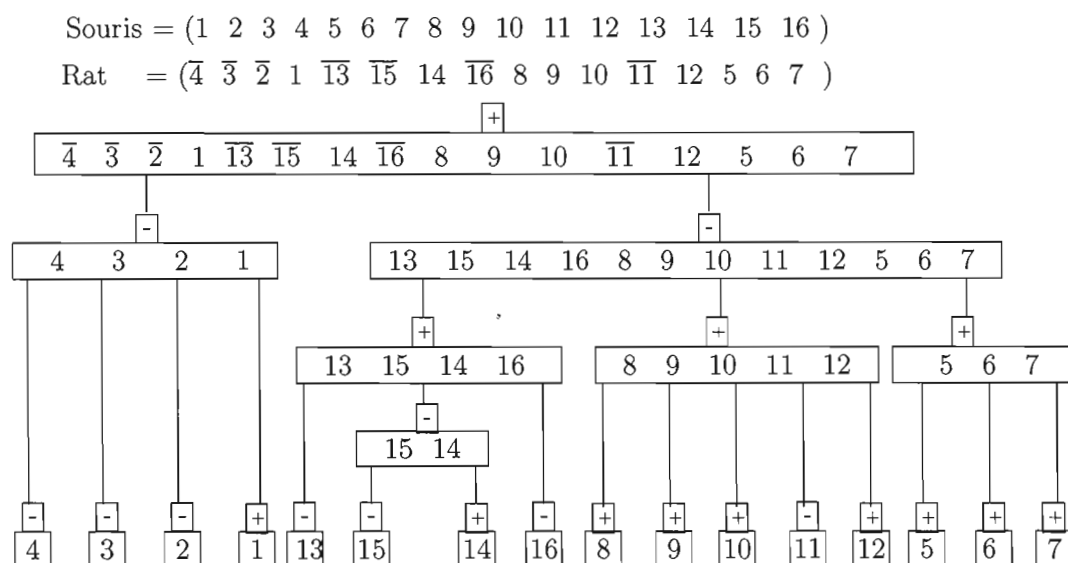
Pour déterminer les trois permutations, il suffit de fixer un des trois chromosomes comme repère, que nous représenterons par la permutation identité. Ensuite, les permutations qui représentent les deux autres espèces sont déterminées en identifiant, pour chaque bloc du chromosome repère, la position et l'orientation de son bloc identique chez chacune des autres espèces.

Lorsqu'on a un nombre négatif,  $\bar{x}$ , chez une espèce, cela signifie que, dans cette espèce, le bloc identique au bloc  $x$  de l'espèce repère se trouve sur le brin opposé. En fixant le chromosome X de l'homme comme repère, la Figure 3.6 illustre les trois permutations obtenues par cette comparaison.

Prenons l'exemple du bloc 1 du chromosome X de l'homme, le bloc correspondant se retrouve à la huitième position chez la souris et à la neuvième position chez le rat, tandis que le bloc 2 de l'homme se retrouve à la seizième position chez la souris mais sur un brin opposé et à la septième position chez le rat (voir Figure 3.6).

Notre intérêt pour ces données est basé sur les arbres des intervalles forts construits à partir des permutations qui comparent ces espèces deux à deux. En effet, la comparaison des 16 blocs de synténie des chromosomes X de la souris et du rat donne un arbre des intervalles forts qui est défini. Cet arbre est présenté dans la Figure 3.7.

Dans la Figure 3.8, nous avons l'arbre des intervalles forts des permutations qui représentent les 16 blocs de synténie des chromosomes X de l'homme et du rat, en fixant le chromosome de l'homme comme repère. Cet arbre est non-ambigu.



**FIG. 3.7** L'arbre des intervalles forts des permutations qui représentent les 16 blocs de synténie des chromosomes X de la souris et du rat. C'est un arbre défini car il ne possède aucun noeud premier.

Enfin, nous avons, dans la Figure 3.9, l'arbre des intervalles forts des permutations qui représentent les 16 blocs de synténie des chromosomes X de l'homme et de la souris, en fixant le chromosome de l'homme comme repère. Cet arbre est ambigu car sa racine est un noeud premier dont le signe est indéfini.

Ces résultats sont d'autant plus intéressants qu'ils sont basés sur des données biologiques réelles. Par conséquent, ces arbres serviront d'exemples dans la section suivante qui traite de l'identification des scénarios parfaits et parcimonieux selon le type d'arbre.

### 3.2 Les scénarios parfaits et parcimonieux

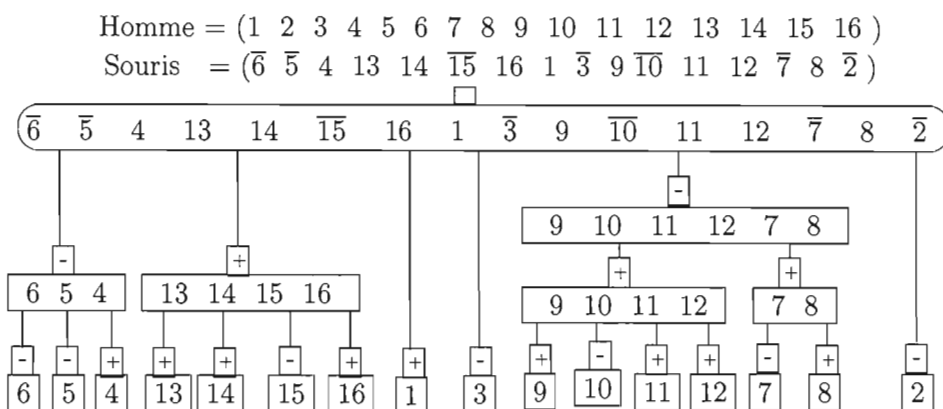
Dans cette section, nous décrirons des algorithmes efficaces qui permettent de calculer des scénarios parfaits et parcimonieux sur de grandes classes de permutations signées. Nous commencerons par identifier les intervalles qui font partie d'un scénario parfait. Ensuite, nous discuterons de la manière d'identifier les scénarios parfaits et parcimonieux selon qu'un arbre est de type défini, non-ambigu ou ambigu.

Nous avons déjà mentionné que certains intervalles ne peuvent pas faire partie d'un scénario parfait car ils brisent des intervalles communs. La Proposition 6 identifie les intervalles qui peuvent appartenir à un scénario parfait.

*Démonstration.* Supposons que  $S$  est un scénario de  $P$  et que toute inversion de  $S$  est soit à un noeud de  $T_f(P)$ , soit à l'union des enfants d'un noeud premier de  $T_f(P)$ .

Supposons, maintenant, que  $I$  est une union des enfants d'un noeud premier  $J$  de l'arbre  $T_f(P)$ . Ce cas est plus complexe que le précédent. Pour le simplifier, nous allons le diviser en petites étapes faciles à comprendre.

D'abord, notons qu'un intervalle commun de  $P$  ne peut pas chevaucher l'intervalle



**FIG. 3.9** L'arbre des intervalles forts construits à partir des permutations qui comparent les chromosomes X de l'homme et de la souris. C'est un arbre ambigu.

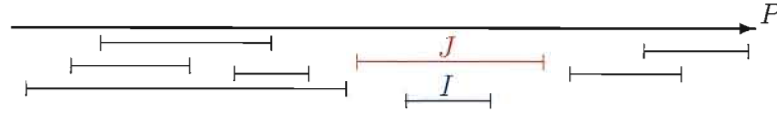
fort  $J$ , par la définition. Ainsi, tout intervalle commun de  $P$  est soit disjoint de  $J$ , soit inclus dans  $J$ . Aussi, un intervalle commun de  $P$  ne peut pas chevaucher un ou plusieurs enfants de  $J$  car les enfants de  $J$  sont des intervalles forts. Par conséquent, parmi les intervalles communs de  $P$  inclus dans  $J$ , nous avons ceux qui sont inclus dans les enfants de  $J$  et ceux qui sont formés par l'union des enfants de  $J$ .

Ainsi, par rapport à l'intervalle fort  $J$ , nous avons trois catégories d'intervalles communs de  $P$ . La première catégorie est composée de ceux qui sont disjoints de l'intervalle  $J$ , la deuxième catégorie est composée par ceux qui sont inclus dans les enfants de  $J$  et enfin, la troisième catégorie est composée par ceux qui sont formés par l'union des enfants de  $J$ .

Nous allons maintenant démontrer que l'intervalle  $I$  commute avec les intervalles communs de  $P$  qui appartiennent à chacune de ces trois catégories.

- a) Commençons par la catégorie des intervalles communs de  $P$  qui sont disjoints de l'intervalle fort  $J$ . Comme  $J$  est disjoint de ces intervalles et que  $I$  est inclus dans  $J$  alors, il est évident que  $I$  est disjoint des intervalles de cette catégorie. La Figure 3.10 illustre clairement ce fait, les intervalles communs de  $P$  disjoints de  $J$  sont dessinés en noir.
- b) Analysons, maintenant, les intervalles communs de  $P$  inclus dans les enfants de  $J$ .

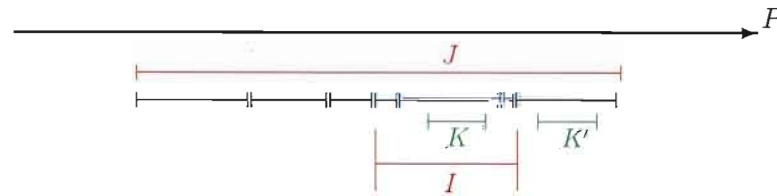




**FIG. 3.10** L'intervalle  $I$  est inclus dans l'intervalle fort  $J$ . Les intervalles communs de  $P$  qui sont disjoints de  $J$  (ils sont dessinés en noir) commutent avec l'intervalle  $I$ .

Rappelons que  $I$  est une union de certains enfants de  $J$ . Ainsi, un enfant de  $J$  est soit inclus dans  $I$ , soit disjoint de  $I$  et dans les deux cas,  $I$  ne peut pas chevaucher un intervalle commun inclus dans cet enfant.

Dans la Figure 3.11, les enfants de  $J$  qui sont inclus dans  $I$  sont dessinés en bleu, tandis que ceux qui sont disjoints de  $I$  sont dessinés en noir. On constate que, si un intervalle commun  $K$  est inclus dans un enfant de  $J$  et que cet enfant est inclus dans  $I$  alors, nécessairement,  $K$  est inclus dans  $I$ . Aussi, si un intervalle commun  $K'$  est inclus dans un enfant disjoint de  $I$  alors, obligatoirement  $K'$  est disjoint de l'intervalle  $I$ .



**FIG. 3.11** L'intervalle  $I$  commute avec tout intervalle commun de  $P$  inclus dans un enfant de l'intervalle  $J$ .

- c) Enfin, l'intervalle  $I$  ne doit chevaucher aucun intervalle commun de  $P$  qui est formé par une union d'enfants de  $J$ , mais par définition d'un noeud premier, un tel intervalle commun n'existe pas.

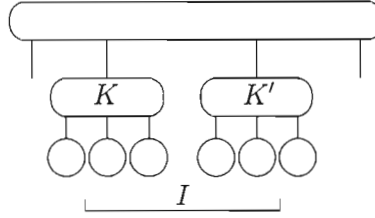
En conclusion, l'intervalle  $I$  commute avec tous les intervalles communs de  $P$ .

Supposons, maintenant, que le scénario  $S$  est parfait. Posons  $I$ , un intervalle contenu dans  $S$  et considérons que la partition  $\{I_1, I_2, \dots, I_k\}$  de  $I$  dans lequel la partie qui contient un élément  $x$  de  $I$  est l'intervalle fort le plus grand contenu dans  $I$  qui

contient  $x$ .

Si  $k \geq 2$ , alors  $I_1, I_2, \dots, I_{k-1}$  et  $I_k$  doivent, tous, être des enfants du même parent  $J$  dans l'arbre  $T_f(P)$ . Sinon  $I$  chevauche les noeuds de  $T_f(P)$  qui sont les parents de quelques intervalles parmi les éléments de la partition  $\{I_1, I_2, \dots, I_k\}$ .

En effet, supposons, par exemple, qu'il existe deux noeuds  $K$  et  $K'$  de l'arbre  $T_f(P)$  qui sont des parents de deux sous-ensembles de la partition  $\{I_1, \dots, I_k\}$ . Comme nous pouvons le voir dans la Figure 3.12, l'intervalle  $I$  chevauche les noeuds  $K$  et  $K'$ .



**FIG. 3.12** Les intervalles qui forment  $I$  sont quelques uns parmi les enfants de deux noeuds différents  $K$  et  $K'$  de l'arbre des intervalles forts. Ainsi,  $I$  chevauche les parents  $K$  et  $K'$ .

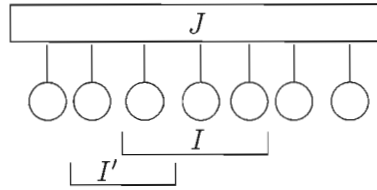
Remarquons que  $I$  ne peut pas contenir les noeuds  $K$  et  $K'$ . En effet, dans ce cas, l'hypothèse selon laquelle l'intervalle  $I_j$  de la partition de  $I$  qui contient l'élément  $x$  est l'intervalle fort le plus grand contenu dans  $I$  qui contient  $x$ , serait violée.

Si  $J$  est un noeud linéaire, alors  $I$  doit être égal à  $J$ . Sinon,  $I$  chevauche l'intervalle commun  $I'$  formé par un des enfants de  $J$  qui n'appartient pas à  $I$  et l'un des intervalles forts contenus dans  $I$ . Or, un tel intervalle est un intervalle commun de  $P$  par les points 1 et 2 du Théorème 1.

En effet, dans la Figure 3.13 par exemple, on observe que l'intervalle  $I$  chevauche l'intervalle  $I'$  qui est un intervalle commun de  $P$ . Ainsi, soit  $k = 1$  et  $I$  est un noeud de  $T_f(P)$  ou  $k > 1$  et le noeud  $J$  doit être premier.

□

Alors que la Proposition 6 identifie les intervalles qui peuvent appartenir à un



**FIG. 3.13** L'intervalle commun  $I'$  est formé par l'union d'un enfant de  $J$  qui n'appartient pas à  $I$  avec un intervalle fort inclus dans  $I$ . Ainsi,  $I$  chevauche  $I'$  et donc inverser l'intervalle  $I$  brise l'intervalle commun  $I'$  de la permutation  $P$ .

scénario parfait, le Lemme 5, ci-dessous, identifie les intervalles qui doivent appartenir à tout scénario parfait, donc au scénario parfait et parcimonieux aussi.

**Lemme 5.** *Soit  $I$  un noeud d'un arbre des intervalles forts  $T_f(P)$  d'une permutation signée  $P$ . Si  $I$  possède un parent linéaire et un signe différent du signe de son parent, alors  $I$  appartient à tout scénario parfait de la permutation  $P$ .*

*Démonstration.* Soit  $S$ , un scénario parfait et  $I$ , un noeud de l'arbre des intervalles forts qui a un signe négatif et dont le parent linéaire  $J$  a un signe positif, tel que  $I \notin S$ .

Notons que  $I$  ne peut pas être un noeud premier. En effet, les noeuds premiers héritent leur signe de leur parent lorsque ce dernier est linéaire et  $I$  possède un signe différent de celui de son parent linéaire  $J$ .

Sachant que  $I \notin S$  et que son parent  $J$  est linéaire, nous en concluons que toute inversion incluse dans  $S$  qui contient  $I$  contient aussi  $J$  selon la Proposition 6.

Soit  $m$ , le nombre d'inversions dans le scénario  $S$  qui contient  $J$ . Si  $m$  est pair, comme  $J$  est de signe positif, il le sera aussi après  $m$  inversions. Ainsi, le scénario  $S$  devra trier la permutation  $P$  vers l'identité, par définition d'un noeud croissant.

Si  $I$  est une feuille, il sera encore négatif après un nombre pair d'inversions, ce qui contredit le fait que  $S$  trie  $P$  vers l'identité. En effet, l'identité ne comprend pas d'élément négatif.

Et si  $I$  est un noeud linéaire avec un signe négatif, alors son premier enfant est plus grand que son dernier enfant et cette situation persistera après un nombre pair d'inversions, ce qui contredit, encore une fois, le fait que  $S$  trie  $P$  vers l'identité.

Un argument symétrique pourrait être fait dans le cas où  $m$  est impair ou si  $I$  est de signe positif et  $J$  est de signe négatif.  $\square$

Nous avons identifié les intervalles qui peuvent appartenir à un scénario parfait dans la Proposition 6. Ensuite, dans le Lemme 5, nous avons vu les intervalles qui doivent appartenir à un scénario parfait. Dans les sections suivantes, nous verrons les algorithmes qui identifient des scénarios parfaits et parcimonieux selon le type de l'arbre d'intervalles forts.

### 3.2.1 Identifier un scénario parfait et parcimonieux avec un arbre des intervalles forts défini

Nous avons déjà mentionné que la façon d'identifier un scénario parfait et parcimonieux d'une permutation signée donnée dépend de la structure de son arbre des intervalles forts. Dans cette section, nous allons discuter de la manière d'identifier un scénario parfait et parcimonieux d'une permutation signée dont l'arbre des intervalles forts est défini.

Remarquez que, lorsque l'arbre des intervalles forts d'une permutation  $P$  est défini alors  $P$  fait partie de la classe des permutations les plus faciles à trier. En effet, le Lemme 5 implique le théorème ci-dessous qui identifie le scénario parfait et parcimonieux de la permutation  $P$  d'une façon efficace et précise.

**Théorème 2.** *Soit  $P$  une permutation signée. Si l'arbre des intervalles forts  $T_f(P)$  de la permutation  $P$  est défini, alors l'ensemble des noeuds qui possèdent un signe différent de celui de leur parent constitue un scénario parfait et parcimonieux de  $P$ . De plus, aucun autre noeud que ceux-ci n'appartient à un tel scénario.*

*Démonstration.* Par la Proposition 6, sachant que l'arbre des intervalles forts défini ne

possède aucun de noeud premier, le scénario parfait est constitué par l'ensemble de noeuds de  $T_f(P)$ . De plus, le Lemme 5 dit que tout scénario parfait et parcimonieux est composé de noeuds de  $T_f(P)$  qui ont un signe différent de celui de leur parent.  $\square$

Ainsi, selon le Théorème 2, le calcul d'un scénario parfait et parcimonieux qui trie une permutation  $P$  vers l'identité (ou l'identité inversée) est presque immédiat lorsque l'arbre des intervalles forts est défini. Il suffit de faire un simple parcours de l'arbre  $T_f(P)$  en comparant les signes de ses noeuds.

**Algorithme 2** : L'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est défini.

**Entrée** : L'arbre des intervalles forts défini  $T_f(P)$  d'une permutation  $P$ .

**Sortie** : Le scénario  $S$  parfait et parcimonieux qui trie  $P$ .

```

1 : Tous les noeuds de l'arbre  $T_f(P)$  sont non visités ;
2 :  $S$  est un scénario vide ;
3 : Parcourir l'arbre  $T_f(P)$  dans l'ordre préfixe ;
4 :    $I \leftarrow$  racine de l'arbre ;
5 :   tant que il existe un noeud non visité dans l'arbre  $T_f(P)$  faire
6 :     si  $((I \neq \text{premier}) \text{ ET } (\text{signe de } I \neq \text{signe de son parent}))$  alors
7 :       Ajouter  $I$  dans  $S$  ;
8 :     fin
9 :      $I$  devient un noeud visité ;
10 :     $I \leftarrow$  noeud suivant ;
11 :  fin
```

Prenons l'exemple de l'arbre de la Figure 3.7. On y voit l'arbre des intervalles forts de la permutation  $P = (\overline{4} \ \overline{3} \ \overline{2} \ 1 \ \overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \overline{11} \ 12 \ 5 \ 6 \ 7)$  qui représente l'ordre des 16 blocs de synténie dans le chromosome X du rat en comparaison avec le chromosome X de la souris. Cet arbre est défini. La Figure 3.14 nous montre le scénario parfait et parcimonieux composé de 11 inversions qui trie  $P$  vers la permutation identité.

$$\begin{aligned}
r_0 : & (\underline{4} \ \underline{3} \ \underline{2} \ 1 \ \overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \overline{11} \ 12 \ 5 \ 6 \ 7) \\
r_1 : & (\underline{1} \ 2 \ 3 \ 4 \ \overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \overline{11} \ 12 \ 5 \ 6 \ 7) \\
r_2 : & (1 \ 2 \ 3 \ 4 \ \underline{\overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \overline{11} \ 12 \ 5 \ 6 \ 7}) \\
r_3 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ \overline{12} \ 11 \ \overline{10} \ \overline{9} \ \overline{8} \ \underline{16 \ \overline{14} \ 15 \ 13}) \\
r_4 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ \overline{12} \ 11 \ \overline{10} \ \overline{9} \ \overline{8} \ \underline{13 \ \overline{15} \ 14 \ \overline{16}}) \\
r_5 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ \overline{12} \ 11 \ \overline{10} \ \overline{9} \ \overline{8} \ 13 \ \underline{\overline{15} \ 14 \ \overline{16}}) \\
r_6 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ \overline{12} \ 11 \ \overline{10} \ \overline{9} \ \overline{8} \ 13 \ \underline{\overline{14}} \ 15 \ \overline{16}) \\
r_7 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ \overline{12} \ 11 \ \overline{10} \ \overline{9} \ \overline{8} \ 13 \ 14 \ 15 \ \underline{\overline{16}}) \\
r_8 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ \underline{\overline{12} \ 11 \ \overline{10} \ \overline{9} \ \overline{8}} \ 13 \ 14 \ 15 \ 16) \\
r_9 : & (1 \ 2 \ 3 \ 4 \ \overline{7} \ \overline{6} \ \overline{5} \ 8 \ 9 \ 10 \ \underline{\overline{11}} \ 12 \ 13 \ 14 \ 15 \ 16) \\
r_{10} : & (1 \ 2 \ 3 \ 4 \ \underline{\overline{7} \ \overline{6} \ \overline{5}} \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16) \\
r_{11} : & (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16)
\end{aligned}$$

**FIG. 3.14** Le scénario  $S$  qui trie  $P$  vers l'identité. Pour représenter l'intervalle qui sera inversé à chaque étape, nous le soulignons. Après chaque inversion, la permutation à la ligne suivante montre la résultante avec la prochaine inversion à effectuer s'il y a lieu.

Notons que, ce scénario est unique à l'ordre près. En effet, étant donné que c'est un arbre défini, toutes les inversions du scénario sont des noeuds de l'arbre des intervalles forts. Or, nous savons, par définition de l'arbre des intervalles forts, que chaque noeud commute avec tous les intervalles communs de  $P$ .

Par conséquent, les inversions de ce scénario peuvent apparaître dans n'importe quel ordre. Par exemple, le scénario de la Figure 3.15 est le même que celui de la Figure 3.14 à l'ordre près.

Cependant, ceci n'est pas toujours le cas. En effet, dans certains scénarios, l'ordre des inversions est fixe et on ne peut pas interchanger les inversions qui compose un scénario à notre guise. Nous verrons de tels scénarios plus loin.

$$\begin{aligned}
r_0 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \underline{1} \ \overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \overline{11} \ 12 \ 5 \ 6 \ 7 ) \\
r_1 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ \overline{13} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ \underline{\overline{11}} \ 12 \ 5 \ 6 \ 7 ) \\
r_2 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ \underline{\overline{13}} \ \overline{15} \ 14 \ \overline{16} \ 8 \ 9 \ 10 \ 11 \ 12 \ 5 \ 6 \ 7 ) \\
r_3 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ 13 \ \overline{15} \ \underline{14} \ \overline{16} \ 8 \ 9 \ 10 \ 11 \ 12 \ 5 \ 6 \ 7 ) \\
r_4 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ 13 \ \overline{15} \ \overline{14} \ \underline{\overline{16}} \ 8 \ 9 \ 10 \ 11 \ 12 \ 5 \ 6 \ 7 ) \\
r_5 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ 13 \ \underline{\overline{15}} \ \underline{\overline{14}} \ 16 \ 8 \ 9 \ 10 \ 11 \ 12 \ 5 \ 6 \ 7 ) \\
r_6 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ \underline{13} \ \underline{14} \ \underline{15} \ \underline{16} \ 8 \ 9 \ 10 \ 11 \ 12 \ 5 \ 6 \ 7 ) \\
r_7 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ \overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \underline{8} \ \underline{9} \ \underline{10} \ \underline{11} \ \underline{12} \ 5 \ 6 \ 7 ) \\
r_8 : & ( \overline{4} \ \overline{3} \ \overline{2} \ \overline{1} \ \overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \overline{12} \ \overline{11} \ \overline{10} \ \overline{9} \ \overline{8} \ \underline{5} \ \underline{6} \ \underline{7} ) \\
r_9 : & ( \underline{\overline{4}} \ \underline{\overline{3}} \ \underline{\overline{2}} \ \underline{\overline{1}} \ \overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \overline{12} \ \overline{11} \ \overline{10} \ \overline{9} \ \overline{8} \ \overline{7} \ \overline{6} \ \overline{5} ) \\
r_{10} : & ( 1 \ 2 \ 3 \ 4 \ \underline{\overline{16}} \ \underline{\overline{15}} \ \underline{\overline{14}} \ \underline{\overline{13}} \ \underline{\overline{12}} \ \underline{\overline{11}} \ \underline{\overline{10}} \ \underline{\overline{9}} \ \underline{\overline{8}} \ \underline{\overline{7}} \ \underline{\overline{6}} \ \underline{\overline{5}} ) \\
r_{11} : & ( 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 )
\end{aligned}$$

**FIG. 3.15** Ce scénario est composé par les mêmes inversions que le scénario de la Figure 3.14 à l'ordre près. Mais, ce changement ne modifie pas les résultats du tri.

### 3.2.2 Un scénario parfait et parcimonieux avec un arbre des intervalles forts non-ambigu

Nous allons maintenant analyser le cas des permutations dont l'arbre des intervalles forts est non-ambigu. Dans un tel arbre, il peut y avoir un ou plusieurs noeuds premiers, cependant tous les parents des noeuds premiers sont linéaires.

Suite à la définition d'un arbre non-ambigu et aux contraintes imposées par l'Algorithme 1, la façon d'affecter des signes aux noeuds d'un arbre des intervalles forts non-ambigu  $T_f(P)$  d'une permutation  $P$  est unique. Ainsi, nous savons exactement dans quel sens nous devons trier chacun des noeuds de  $T_f(P)$ . Le théorème ci-dessous identifie un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est non-ambigu.

**Théorème 3.** *Soit une permutation signée  $P$  dont l'arbre des intervalles forts  $T_f(P)$*

est non-ambigu. Un scénario  $S$  parfait et parcimonieux de  $P$  est identifié en deux étapes distinctes :

- 1) à l'étape  $A$ , on ajoute à  $S$  les intervalles qui ordonnent les enfants des noeuds premiers de l'arbre  $T_f(P)$  selon leur signe d'une façon parcimonieuse et
- 2) à l'étape  $B$ , on ajoute à  $S$  les intervalles correspondant aux noeuds linéaires ou aux feuilles qui possèdent un signe différent de celui de leur parent linéaire.

*Démonstration.* La démonstration comporte deux parties. D'abord nous allons démontrer qu'un scénario composé par les intervalles énumérées dans ce théorème est parfait et ensuite, nous allons prouver que, parmi les scénarios parfaits qui trient cette permutation, le scénario identifié est le plus parcimonieux.

Par définition, il n'existe aucun ordre naturel sur les enfants d'un noeud premier. Ainsi, si on ordonne les enfants d'un noeud premier de l'arbre  $T_f(P)$  selon son signe, il perd sa qualité de noeud premier et il devient un noeud linéaire.

Pour ordonner les enfants d'un noeud premier, il suffit de trier sa permutation-quotient. Notons que, durant cette opération, on n'a pas à se soucier de la sauvegarde des intervalles communs car ils n'existent pas. Ainsi, la seule contrainte appliquée à ce tri est la parcimonie. Il existe plusieurs algorithmes que nous pouvons utiliser pour ordonner les enfants des noeuds premiers de façon parcimonieuse [Tannier et al.(20), Bergeron et al. (3), etc.].

Ainsi, après avoir ordonné les enfants de tous les noeuds premiers de l'arbre  $T_f(P)$  dans l'étape  $A$ , le problème devient celui d'identifier un scénario parfait et parcimonieux avec un arbre défini, car tous les noeuds premiers contenus dans l'arbre  $T_f(P)$  de départ sont maintenant des noeuds linéaires. Ainsi, à l'étape  $B$ , nous appliquons la méthode décrite dans la section précédente.

Comme on ne brise aucun intervalle commun de  $P$  ni lorsqu'on ordonne les enfants des noeuds premiers de  $T_f(P)$ , ni lorsqu'on inverse un noeud de l'arbre  $T_f(P)$  qui possède un signe opposé à celui de son parent linéaire alors, un scénario constitué par ces deux



catégories d'intervalles est nécessairement parfait.

Ensuite, il faut démontrer que le scénario parfait obtenu par la méthode décrite dans ce théorème, appelons-le  $S$ , est le plus parcimonieux de tous les scénarios parfaits possibles.

La séquence d'inversions qui sont contenues dans un noeud premier  $I$  trie le noeud  $I$  en ordre croissant ou décroissant selon le signe de  $I$ .

Supposons que  $S'$  est un scénario parcimonieux plus court que le scénario  $S$ . Alors, il doit y avoir, au moins, un noeud premier  $I$  tel que le nombre d'inversions du scénario  $S'$  qui sont contenues dans  $I$  est moins élevé que le nombre d'inversions du scénario  $S$  qui sont contenues dans  $I$ .

Posons  $R$  et  $R'$ , les séquences d'inversions respectives de  $S$  et de  $S'$  qui sont contenues dans  $I$ . L'une de ces deux séquences trie  $I$  dans l'ordre croissant et l'autre dans l'ordre décroissant. Sinon, l'un parmi eux ne pourra pas être parcimonieux.

Supposons, maintenant, que le parent du noeud  $I$  est de signe positif, alors  $R$  trie  $I$  en ordre croissant et  $R'$  en ordre décroissant. Par le même argument que nous avons utilisé pour le Lemme 5,  $I$  doit appartenir au scénario  $S'$  et aussi à la séquence d'inversions  $R'$ . De plus, enlever  $I$  du scénario  $S'$  produit un scénario plus court, ce qui contredit l'hypothèse que  $S'$  était parcimonieux. Un argument similaire peut être utilisé pour justifier le cas où le parent du noeud  $I$  est de signe négatif.

□

Nous avons un exemple d'un arbre des intervalles forts non-ambigu dans la Figure 3.8 qui montre l'arbre associé aux permutations qui représentent les 16 blocs de synténie des chromosomes X de l'homme et du rat. Le noeud (4 5 6 12 8 7 2 1 3 9 10 11) de cet arbre est premier mais son parent est linéaire et décroissant. Ainsi, il hérite le signe négatif de son parent, d'où il faudra le trier vers l'identité inversée.

La permutation-quotient associée à ce noeud est  $P_I = (2 \bar{5} \bar{3} 1 4)$ . La Figure 3.16 montre l'effet de chacune des inversions qui constituent un scénario parcimonieux qui trie  $P_I$  (voir Annexe A.1), puis son effet correspondant sur la permutation  $P$ .

Scénario de $P_I = (2 \bar{5} \bar{3} 1 4)$	Effet sur $P = (\bar{13} \bar{4} 5 \bar{6} \bar{12} \bar{8} \bar{7} 2 1 \bar{3} 9 10 11 14 \bar{15} 16)$
$(2 \bar{5} \bar{3} 1 4)$	$(\bar{13} \bar{4} 5 \bar{6} \bar{12} \bar{8} \bar{7} 2 1 \bar{3} 9 10 11 14 \bar{15} 16)$
$(2 \bar{5} \bar{4} \bar{1} 3)$	$(\bar{13} \bar{4} 5 \bar{6} \bar{12} \bar{11} \bar{10} \bar{9} \bar{3} \bar{1} \bar{2} 7 8 14 \bar{15} 16)$
$(2 \bar{5} \bar{4} \bar{3} \bar{1})$	$(\bar{13} \bar{4} 5 \bar{6} \bar{12} \bar{11} \bar{10} \bar{9} \bar{8} \bar{7} 2 1 \bar{3} 14 \bar{15} 16)$
$(2 \bar{5} \bar{4} \bar{3} \bar{1})$	$(\bar{13} \bar{4} 5 \bar{6} \bar{12} \bar{11} \bar{10} \bar{9} \bar{8} \bar{7} 3 \bar{1} \bar{2} 14 \bar{15} 16)$
$(3 \bar{4} \bar{5} \bar{2} \bar{1})$	$(\bar{13} \bar{7} \bar{8} \bar{9} 10 11 \bar{12} 6 \bar{5} 4 3 \bar{1} \bar{2} 14 \bar{15} 16)$
$(\bar{5} \bar{4} \bar{3} \bar{2} \bar{1})$	$(\bar{13} \bar{12} \bar{11} \bar{10} \bar{9} \bar{8} \bar{7} 6 \bar{5} 4 3 \bar{1} \bar{2} 14 \bar{15} 16)$

**FIG. 3.16** Le scénario parcimonieux qui trie la permutation-quotient  $(2 \bar{5} \bar{3} 1 4)$  vers  $\bar{Id}_n$  et l'effet correspondant à chacune de ses inversions sur la permutation de départ.

Remarquer que, contrairement aux inversions effectuées à l'étape  $B$  qui peuvent être exécutées dans n'importe quel ordre, les inversions comprises dans les scénarios parcimonieux qui trient les permutations-quotients des noeuds premiers dans l'étape  $A$  doivent respecter l'ordre.

En effet, si on exécute les inversions du scénario de la Figure 3.16 qui trie la permutation  $P_I = (2 \bar{5} \bar{3} 1 4)$  dans n'importe quel ordre, cela ne fonctionne pas nécessairement. Par exemple, l'inversion  $\{2, 3, 4, 5\}$  fait partie de ce scénario. Cependant, on ne peut pas l'exécuter en premier car elle ne représente pas un intervalle dans  $P_I$ .

En partant de la permutation qu'on obtient après que la permutation  $P$  ait subi les 5 inversions qui trie  $P_I$ , soit la permutation  $(\bar{13} \bar{12} \bar{11} \bar{10} \bar{9} \bar{8} \bar{7} 6 \bar{5} 4 3 \bar{1} \bar{2} 14 \bar{15} 16)$ , nous allons appliquer l'étape  $B$  de l'Algorithme 3 pour arriver à la permutation identité. Le scénario résultant cette étape est représenté dans la Figure 3.17.

Ainsi, au total, un scénario parfait et parcimonieux qui trie la permutation signée

$(\overline{13} \ \overline{12} \ \overline{11} \ \overline{10} \ \overline{9} \ \overline{8} \ \overline{7} \ 6 \ 5 \ 4 \ 3 \ \underline{1} \ \underline{2} \ 14 \ \overline{15} \ 16)$   
 $(2 \ 1 \ \overline{3} \ \underline{4} \ 5 \ \overline{6} \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \overline{15} \ 16)$   
 $(2 \ 1 \ \overline{3} \ 4 \ 5 \ \underline{6} \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \overline{15} \ 16)$   
 $(\underline{2} \ \underline{1} \ \overline{3} \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \overline{15} \ 16)$   
 $(\underline{1} \ \underline{2} \ \overline{3} \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \overline{15} \ 16)$   
 $(1 \ \underline{2} \ \overline{3} \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \overline{15} \ 16)$   
 $(1 \ 2 \ \underline{3} \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \overline{15} \ 16)$   
 $(1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ \underline{\overline{15}} \ 16)$   
 $(1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16)$

**FIG. 3.17** Ce scénario complète celui de la Figure 3.16. Il illustre l'étape  $B$  de l'Algorithme 3.

$P = (\overline{13} \ \overline{4} \ 5 \ \overline{6} \ \overline{12} \ \overline{8} \ \overline{7} \ 2 \ 1 \ \overline{3} \ 9 \ 10 \ 11 \ 14 \ \overline{15} \ 16)$  vers l'identité comporte 13 inversions soit 5 inversions effectuées à l'étape  $A$  et 8 inversions à l'étape  $B$ .

L'Algorithme 3 implémente la méthode de calcul d'un scénario parfait et parcimonieux décrite dans le Théorème 3.

### 3.2.3 Un scénario parfait et parcimonieux avec un arbre des intervalles forts ambigu

Un arbre des intervalles forts  $T_f(P)$  d'une permutation  $P$  est ambigu lorsqu'il possède au moins un noeud premier de signe indéfini. Ces permutations sont les plus difficiles à trier.

En effet, comme nous l'avons déjà mentionné plus haut, lorsqu'un noeud premier  $I$  est de signe indéfini, cela signifie qu'il n'y a aucune information qui nous aide à déterminer dans quel ordre  $I$  devrait être trié. Dans un tel cas, on essaie toutes les possibilités puis on choisit le meilleur résultat.

**Définition 15.** Un arbre des intervalles forts  $T_f(P)$  d'une permutation  $P$  est *proprement signé* si aucun noeud de l'arbre  $T_f(P)$  n'est de signe indéfini. Les arbres non-ambigus

**Algorithme 3** : L'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est non-ambigu.

**Entrée** : L'arbre des intervalles forts non-ambigu  $T_f(P)$  d'une permutation  $P$ .

**Sortie** : Le scénario  $S$  parfait et parcimonieux qui trie  $P$ .

```

1 :  $S$  est un scénario vide ;
2 : pour chaque noeud premier  $I$  de  $T_f(P)$  faire
3 :    $P_I$  est la permutation-quotient de  $I$  sur ses enfants ;
4 :   si le signe de  $I$  est positif alors
5 :     calculer un scénario parcimonieux  $T$  de  $P_I$  vers  $Id$  ;
6 :   sinon
7 :     calculer un scénario parcimonieux  $T$  de  $P_I$  vers  $\overline{Id}$  ;
8 :   fin
9 :   déduire le scénario correspondant  $T'$  sur les enfants de  $I$  ;
10 :  ajouter les inversions du scénario  $T'$  au scénario  $S$  ;
11 : fin
12 : appliquer l'Algorithme 2 ;

```

sont toujours proprement signés.

**Théorème 4.** *Soit une permutation signée  $P$ , dont l'arbre des intervalles forts  $T_f(P)$  est ambigu avec  $k$  noeuds premiers de signe indéfini. Le calcul d'un scénario parfait et parcimonieux de  $P$  consiste en deux étapes suivantes :*

- 1) *à calculer un scénario parfait et parcimonieux sur chacun des  $2^k$  arbres des intervalles forts proprement signés qu'on obtient en accordant un signe aux  $k$  noeuds ;*
- 2) *à choisir le scénario le plus parcimonieux de tous, ce qui représente la solution au problème.*

*Démonstration.* D'abord, il faut transformer l'arbre  $T_f(P)$  en arbres proprement signés. Sachant que nous avons  $k$  noeuds de signe indéfini et que chaque noeud peut soit le signe positif, soit le signe négatif, nous aurons :

$$\underbrace{2 \times 2 \times \dots \times 2}_k = 2^k \text{ possibilités, donc } 2^k \text{ arbres proprement signés.}$$

Maintenant, il faut calculer un scénario parfait et parcimonieux associé à chacun de ces arbres. Pour ce faire, nous pouvons utiliser la méthode décrite dans la section précédente pour les arbres non-ambigus. Une fois les  $2^k$  scénarios calculés, il reste à déterminer lequel est la solution.

Notons que la différence entre les  $2^k$  scénarios calculés se trouve dans la séquence qui trie parcimonieusement les  $k$  noeuds premiers de signe indéfini. En effet, la séquence qui contient les noeuds linéaires et les feuilles dont le signe est différent de celui de leur parent linéaire est la même dans tous ces  $2^k$  scénarios.

Ainsi, trouver le meilleur scénario revient à déterminer le scénario le plus parcimonieux pour trier les  $k$  noeuds premiers de signe indéfini. Il suffit de comparer les longueurs des scénarios. Il arrive qu'on trouve plusieurs scénarios parcimonieux. Dans ce cas, on choisit un parmi eux au hasard qui devient le scénario solutionnaire.

□

L'Algorithme 4 implémente le calcul d'un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts comprend  $k$  noeuds premiers de signe indéfini tel que décrit dans le Théorème 4. Il est une généralisation de l'Algorithme 3 utilisé dans le calcul d'un scénario parfait et parcimonieux lorsque l'arbre des intervalles forts est non-ambigu.

Prenons l'exemple, de l'arbre des intervalles forts qui compare les blocs de synthèse du chromosomes X de l'homme et de la souris représenté dans la Figure 3.9. Cet arbre est ambigu car, son unique noeud premier est la racine.

La permutation-quotient associée à la racine de cet arbre est  $P_I = (\bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2})$ . La longueur minimale du scénario qui trie  $P_I$  vers l'identité est la même que celle du scénario

**Algorithme 4** : L'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation dont l'arbre des intervalles forts est ambigu.

**Entrée** : L'arbre des intervalles forts ambigu  $T_f(P)$  d'une permutation  $P$ .

**Sortie** : Le scénario  $S$  parfait et parcimonieux qui trie  $P$ .

```

1 : Posons  $I_1, \dots, I_k$ , les noeuds premiers de  $T_f(P)$  de signe indéfini ;
2 : pour chaque mot binaire  $W$  de longueur  $k$  faire
3 :   si  $W[j] = 1$  alors
4 :     Affecter à  $I_j$  le signe + ;
5 :   sinon
6 :     Affecter à  $I_j$  le signe - ;
7 :   fin
8 :   Appliquer l'Algorithme 3 sur l'arbre signé résultant ;
9 : fin pour chaque
```

qui la trie vers l'identité inversée, soit 6 inversions (voir Annexe A.2). Dans un tel cas, nous décidons au hasard l'ordre dans lequel nous allons trier la permutation-quotient car les deux scénarios ont la même longueur.

La Figure 3.18 montre un scénario qui trie  $P_I$  vers l'identité inversée et l'effet de chacune de ces inversions sur la permutation  $P$  de départ.

Il faut ajouter au scénario de la Figure 3.18 les noeuds linéaires et les feuilles de l'arbre de la Figure 3.9 qui ont des signes différents de ceux de leur parent linéaire pour compléter un scénario parfait et parcimonieux de  $P$ . La Figure 3.19 montre la suite de ce scénario.

Selon la Figure 3.18, le scénario parcimonieux qui trie la racine de l'arbre des intervalles forts de  $P = (\overline{6} \ \overline{5} \ 4 \ 13 \ 14 \ \overline{15} \ 16 \ 1 \ \overline{3} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{2})$  compte six inversions. Pour le compléter, nous avons effectué six autres inversions dans la Figure 3.19. Donc, la longueur du scénario parfait et parcimonieux qui trie  $P$  vers l' $\overline{Id_n}$  (ou  $Id_n$ ) est 12.

$$\begin{array}{lcl}
P_I = (\overline{4} \ 6 \ 1 \ \overline{3} \ \overline{5} \ \overline{2}) & & P = (\overline{6} \ \overline{5} \ 4 \ 13 \ 14 \ \overline{15} \ 16 \ 1 \ \overline{3} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{2}) \\
\\
(\overline{4} \ 6 \ 1 \ \overline{3} \ \overline{5} \ \overline{2}) & \longrightarrow & (\overline{6} \ \overline{5} \ 4 \ 13 \ 14 \ \overline{15} \ 16 \ 1 \ \overline{3} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{2}) \\
(\underline{4} \ 6 \ 1 \ \overline{3} \ \overline{5} \ \overline{2}) & \longrightarrow & (\underline{4} \ 5 \ 6 \ 13 \ 14 \ \overline{15} \ \underline{16} \ 1 \ \overline{3} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{2}) \\
(\overline{6} \ \overline{4} \ 1 \ \underline{\overline{3} \ \overline{5} \ \overline{2}}) & \longrightarrow & (\overline{16} \ 15 \ \overline{14} \ \overline{13} \ \overline{6} \ \overline{5} \ 4 \ 1 \ \underline{\overline{3} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{2}}) \\
(\overline{6} \ \underline{\overline{4} \ 1 \ 2 \ 5 \ 3}) & \longrightarrow & (\overline{16} \ 15 \ \overline{14} \ \overline{13} \ \underline{\overline{6} \ \overline{5} \ 4 \ 1 \ 2 \ \overline{8} \ 7 \ \overline{12} \ \overline{11} \ 10 \ \overline{9} \ 3}) \\
(\overline{6} \ \underline{\overline{2} \ \overline{1} \ 4 \ 5 \ 3}) & \longrightarrow & (\overline{16} \ 15 \ \overline{14} \ \overline{13} \ \underline{\overline{2} \ \overline{1} \ 4 \ 5 \ 6 \ \overline{8} \ 7 \ \overline{12} \ \overline{11} \ 10 \ \overline{9} \ 3}) \\
(\overline{6} \ \overline{5} \ \overline{4} \ \underline{1 \ 2 \ 3}) & \longrightarrow & (\overline{16} \ 15 \ \overline{14} \ \overline{13} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ 4 \ \underline{1 \ 2 \ 3}) \\
(\overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1}) & \longrightarrow & (\overline{16} \ 15 \ \overline{14} \ \overline{13} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ 4 \ \overline{3} \ \overline{2} \ \overline{1})
\end{array}$$

**FIG. 3.18** Le scénario qui trie la permutation-quotient  $(\overline{4} \ 6 \ 1 \ \overline{3} \ \overline{5} \ \overline{2})$  sur le noeud racine de l'arbre de la Figure 3.9 vers  $\overline{1d}$  et son effet sur la permutation de départ.

Remarquer que, dans l'exemple précédent, les deux scénarios qui trient la permutation-quotient associée au noeud premier de signe indéfini ont la même longueur. Ceci n'est pas toujours le cas.

Dans la Figure 3.20, nous montrons l'exemple de l'arbre des intervalles forts de la permutation  $P = (2 \ 4 \ 1 \ 3)$ . Dans cette permutation, le scénario qui trie  $P$  vers l'identité est plus long que celui qui la trie vers l'identité inversée.

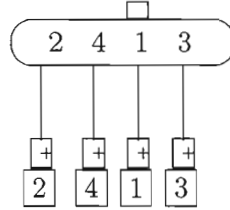
Alors que le scénario parcimonieux qui trie la permutation  $P = (2 \ 4 \ 1 \ 3)$  vers l'identité est de longueur 5, le scénario parcimonieux qui la trie vers l'identité inversée est de longueur 4, comme on peut le voir dans la Figure 3.21. Par conséquent, nous allons choisir de trier cette permutation vers l'identité inversée.

### 3.3 La complexité des algorithmes de calcul des scénarios parfaits et parcimonieux

La complexité de l'algorithme qui calcule un scénario parfait et parcimonieux d'une permutation signée  $P$  dépend du type de l'arbre des intervalles forts de  $P$ . Pour étudier

$(\overline{16} \ 15 \ \overline{14} \ \overline{13} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ \underline{4} \ \overline{3} \ \overline{2} \ \overline{1})$   
 $(\overline{16} \ \underline{15} \ \overline{14} \ \overline{13} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$   
 $(\overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ 9 \ \underline{\overline{10}} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$   
 $(\overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \underline{9 \ 10 \ 11 \ 12} \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$   
 $(\overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \overline{12} \ \overline{11} \ \overline{10} \ \underline{\overline{9}} \ 8 \ \overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$   
 $(\overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \overline{12} \ \overline{11} \ \overline{10} \ \overline{9} \ \underline{7 \ 8} \ \overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$   
 $(\overline{16} \ \overline{15} \ \overline{14} \ \overline{13} \ \overline{12} \ \overline{11} \ \overline{10} \ \overline{9} \ \overline{8} \ \overline{7} \ \overline{6} \ \overline{5} \ \overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$

**FIG. 3.19** Un scénario qui trie la permutation résultante du scénario de la Figure 3.18, soit la permutation  $(\overline{16} \ 15 \ \overline{14} \ \overline{13} \ 9 \ \overline{10} \ 11 \ 12 \ \overline{7} \ 8 \ \overline{6} \ \overline{5} \ 4 \ \overline{3} \ \overline{2} \ \overline{1})$  vers  $\overline{1d}$ .



**FIG. 3.20** L'arbre des intervalles forts de la permutation  $P = (2 \ 4 \ 1 \ 3)$  est un autre exemple où on doit trier un noeud dans les deux sens pour savoir lequel donne le meilleur scénario.

cette complexité, nous allons revoir chacune des étapes nécessaires dans le calcul d'un scénario parfait et parcimonieux d'une permutation signée  $P$ .

Dans la section précédente, nous avons vu les différents algorithmes qui calculent un scénario parfait et parcimonieux de la permutation  $P$  selon que l'arbre  $T_f(P)$  des intervalles forts de  $P$  est défini (Algorithme 2), non-ambigu (Algorithme 3) ou ambigu (Algorithme 4).

L'Algorithme 5 récapitule les différentes étapes vues et expliquées dans ces algorithmes. Dans la suite de cette section, nous analysons les différentes étapes de cet algorithme une à une.

La première étape (ligne 1 de l'Algorithme 5) consiste à construire l'arbre des intervalles forts de la permutation qu'on veut trier. Nous avons déjà mentionné, plus haut, que l'algorithme décrit dans l'article de Bergeron et al. (3) construit un tel arbre



$P$ vers $Id_n$	$P$ vers $\overline{Id}_n$
$(\underline{2} \ 4 \ 1 \ 3)$	$(2 \ \underline{4} \ 1 \ 3)$
$(\overline{2} \ 4 \ \underline{1} \ 3)$	$(2 \ \overline{4} \ \underline{1} \ 3)$
$(\underline{\overline{2}} \ \underline{4} \ \overline{1} \ 3)$	$(2 \ \underline{\overline{4}} \ \underline{\overline{3}} \ \overline{1})$
$(1 \ \overline{4} \ \underline{2} \ \underline{3})$	$(\underline{2} \ \underline{3} \ \underline{4} \ \overline{1})$
$(1 \ \underline{\overline{4}} \ \underline{\overline{3}} \ \underline{\overline{3}})$	$(\overline{4} \ \overline{3} \ \overline{2} \ \overline{1})$
$(1 \ 2 \ 3 \ 4)$	

**FIG. 3.21** Deux scénarios parcimonieux qui trient la permutation  $(2 \ 4 \ 1 \ 3)$  l'un vers  $Id$  et l'autre vers  $\overline{Id}$ . Nous gardons le scénario vers  $\overline{Id}$  car il est le plus court des deux.

dans un temps linéaire par rapport au nombre d'éléments de la permutation.

La deuxième étape (ligne 2 de l'Algorithme 5) consiste à parcourir l'arbre  $T_f(P)$  en accordant un signe à chacun des noeuds selon l'Algorithme 1. Cette étape se fait aussi dans un temps  $O(n)$  comme cela a été expliqué dans la section 3.1.1.

La première et la deuxième étapes doivent être exécutées indépendamment du type de l'arbre des intervalles forts de la permutation. Ensuite, l'algorithme vérifie le type d'arbre, puis il applique l'algorithme approprié. Ainsi, pour analyser la complexité des étapes suivantes, nous le ferons selon le type d'arbre identifié.

### 3.3.1 La complexité de l'algorithme de calcul d'un scénario parfait et parcimonieux avec un arbre défini

Lorsque l'arbre des intervalles forts de  $P$  est défini (ligne 11 de l'Algorithme 5), on applique l'Algorithme 2. Sa complexité est donné par le Corollaire 1.

**Corollaire 1.** *Soit  $P$ , une permutation signée sur  $n$  éléments. Si l'arbre  $T_f(P)$  des intervalles forts de  $P$  est défini, alors le calcul du scénario parfait et parcimonieux de  $P$  se fait en temps linéaire.*

*Démonstration.* Selon le Théorème 2, le calcul du scénario parfait et parcimonieux d'un

**Algorithme 5** : Calcul d'un scénario parfait et parcimonieux de  $P$ .

**Entrée** : La permutation signée  $P$  sur  $n$  éléments.

**Sortie** : Un scénario parfait et parcimonieux  $S$  qui trie  $P$  vers  $Id_n$  ou  $\overline{Id_n}$ .

```

1 : construire l'arbre des intervalles forts  $T_f(P)$  de la permutation  $P$  ;
2 : accorder des signes aux noeuds de l'arbre  $T_f(P)$  ;
3 :  $k \leftarrow$  nombre de noeuds premiers de signe indéfini ;
4 :  $k' \leftarrow$  nombre des noeuds premiers munis d'un signe + ou - ;
5 : si ( $k > 0$ ) alors
6 :     appliquer l'Algorithme 4 ;
7 : sinon si ( $k' > 0$ ) alors
8 :     appliquer l'Algorithme 3;
9 : sinon
10 :     appliquer l'Algorithme 2;
11 : fin
```

arbre défini consiste en un simple parcours de l'arbre des intervalles forts en comparant les signes des noeuds.

Ce parcours prend un temps linéaire par rapport au nombre des noeuds de l'arbre des intervalles forts. Or, par la Proposition 3, nous savons que l'arbre des intervalles forts d'une permutation de  $n$  éléments contient  $(2n - 1)$  noeuds au maximum. Nous en concluons donc, que le scénario parfait et parcimonieux de la permutation  $P$  se calcule en temps  $O(n)$ .  $\square$

### 3.3.2 La complexité de l'algorithme de calcul d'un scénario parfait et parcimonieux avec un arbre non-ambigu

Dans le cas où l'arbre des intervalles forts de la permutation  $P$  est non-ambigu mais qu'il n'est pas défini (ligne 9 de l'Algorithme 5), nous utilisons l'Algorithme 3 pour calculer le scénario parfait et parcimonieux de  $P$ .

Nous avons déjà vu que cet algorithme comprend deux étapes, soit l'étape  $A$  qui calcule un scénario parcimonieux sur la permutation-quotient de chaque noeud premier selon son signe et l'étape  $B$  qui ajoute au scénario de l'étape  $A$  les noeuds de l'arbre  $T_f(P)$  qui possèdent un signe opposé à celui de leur parent linéaire.

Ainsi, la complexité de l'Algorithme 3 dépend de la méthode utilisée à l'étape  $A$  pour calculer les scénarios parcimonieux sur les permutations-quotients des noeuds premiers compris de l'arbre  $T_f(P)$ . L'utilisation de l'algorithme décrit dans l'article de Tannier et al.(20) semble être un bon choix. En effet, dans cet article, les auteurs donnent un algorithme de complexité  $O(n\sqrt{n \log(n)})$ , soit la meilleure complexité connue à l'heure actuelle.

Cependant, malgré sa bonne complexité théorique, l'algorithme de Tannier et al. (20) est complexe et difficile à implémenter. Ainsi, dans l'implémentation que nous avons fait du calcul d'un scénario parfait et parcimonieux, nous avons choisi d'implémenter un autre algorithme basé sur l'article de Bergeron et al. (4), dont la complexité est plus élevée, soit  $O(n^2)$ , mais qui est plus simple à comprendre et plus facile à implémenter.

Le Théorème 5 décrit la complexité de l'Algorithme 3 lorsqu'on utilise l'algorithme de Tannier et al.(20).

**Théorème 5.** *Soit  $P$ , une permutation signée sur  $n$  éléments. Si l'arbre des intervalles forts de  $P$  est non-ambigu, l'Algorithme 3 calcule le scénario parfait et parcimonieux de  $P$  dans un temps dans  $O(n\sqrt{n \log(n)})$ .*

*Démonstration.* Supposons que la permutation  $P$  contient  $k$  noeuds premiers. Chaque noeud premier,  $S_i$  tel que  $1 \leq i \leq k$ , possède un nombre  $n_i$  d'enfants. L'algorithme qui calcule le scénario parcimonieux sur les permutations-quotients des noeuds premiers de  $T_f(P)$  sera appliqué, successivement, sur des permutations de longueur  $n_1, \dots, n_k$ .

Selon l'algorithme de Tannier et al. (20), la complexité de l'étape  $A$  de l'Algorithme 3 est égale à :  $n_1\sqrt{n_1 \log(n_1)} + \dots + n_k\sqrt{n_k \log(n_k)} \leq (n_1 + \dots + n_k)\sqrt{n \log(n)}$  car,  $\forall i$ , on a  $n_i \leq n$ . Aussi, d'après la Proposition 2, nous savons que  $(n_1 + \dots + n_k) \leq n$ .

Ainsi, on peut affirmer que  $(n_1 + \dots + n_k)\sqrt{n \log(n)} \leq (n\sqrt{n \log(n)})$ . Par conséquent, la complexité de cette étape est de l'ordre de  $O(n\sqrt{n \log(n)})$ .

□

### 3.3.3 La complexité de l'algorithme de calcul d'un scénario parfait et parcimonieux avec un arbre ambigu

Dans le cas d'une permutation  $P$  dont l'arbre des intervalles forts est ambigu. Le Théorème 6 donne la complexité du calcul du scénario parfait et parcimonieux de la permutation  $P$ .

**Théorème 6.** *Soit  $P$ , une permutation signée sur  $n$  éléments. Si l'arbre des intervalles forts de la permutation  $P$  est ambigu avec  $k$  noeuds premiers de signe indéfini, alors l'Algorithme calcule le scénario parfait et parcimonieux de  $P$  en  $O(2^k \times n\sqrt{n \log(n)})$ .*

*Démonstration.* Nous avons déjà vu, avec le Théorème 4, que lorsque l'arbre des intervalles forts d'une permutation  $P$  contient un nombre  $k$  de noeuds de signe indéfini, nous devons calculer  $2^k$  scénarios liés aux  $2^k$  arbres proprement signés obtenu à partir de  $T_f(P)$ .

Aussi, nous savons, par ce même théorème, que pour calculer un scénario parfait et parcimonieux sur chacun de ces arbres, on utilise l'Algorithme 3 qui est de complexité  $O(n\sqrt{n \log(n)})$  comme on vient de le démontrer. Ce qui nous donne, en tout,  $O(2^k \times n\sqrt{n \log(n)})$ .

Enfin, il faut identifier le plus parcimonieux des  $2^k$  scénarios. Cependant, cette dernière opération est négligeable comparée à la précédente.

Ainsi, la complexité de l'Algorithme 4 qui calcule le scénario parfait et parcimonieux lorsque l'arbre des intervalles forts est ambigu avec  $k$  noeuds de signe indéfini est de  $O(2^k \times n\sqrt{n \log(n)})$

□

## CONCLUSION

D'un point de vue algorithmique, l'aspect central de ce travail repose sur le lien qui existe entre le calcul d'un scénario parfait et parcimonieux d'une permutation signée quelconque et l'arbre des intervalles forts de cette permutation.

Dans l'article de Figeac et Varré (11), les auteurs introduisent le problème de tri des permutations signées par inversions en conservant les intervalles communs aux permutations. Malheureusement, ils déclarent le problème NP-difficile.

Bérard et al. (2) démontrent qu'il existe une grande classe de permutations pour laquelle on peut calculer un scénario parfait et parcimonieux en temps polynomial et même en temps linéaire dans certains cas. Ce mémoire est une présentation complète des résultats de cet article. Nous avons aussi implanté les algorithmes simples et efficaces qui calculent des scénarios parfaits et parcimonieux.

Nous avons expliqué en détail le rôle primordial joué par l'arbre des intervalles forts d'une permutation signée (Bergeron et al., (4)) dans le calcul d'un scénario parfait et parcimonieux de cette permutation.

Pour terminer, nous allons présenter, brièvement, quelques-unes des études qui viennent compléter le sujet discuté dans ce travail.

Rappelons que l'idée de calculer un scénario qui ne brise pas les structures combinatoires communes aux génomes à comparer repose sur l'hypothèse qu'il y a une raison biologique qui explique l'existence de telles structures. Cependant, ceci n'est pas toujours vrai car un groupe de gènes voisins peut se retrouver dans deux génomes différents juste par hasard. Il serait donc justifié de modifier le problème pour ne conserver qu'un sous-ensemble des intervalles communs dans la recherche du scénario parfait au lieu de

tous les intervalles communs. Dans l'article de Bérard et al. (2), les auteurs discutent de cet aspect du problème.

D'autre part, le but de ce travail est d'identifier un scénario parfait et parcimonieux entre deux génomes seulement. Ainsi, il serait intéressant de considérer le cas de plusieurs espèces. Les auteurs Bernt et al. (5) discutent justement de cet aspect du problème. Dans leur article, ils utilisent le tri parfait et l'arbre des intervalles forts pour résoudre le problème du médian.

Enfin, il existe d'autres notions que les intervalles communs pour modéliser les structures conservées entre deux génomes. Par exemple, dans l'article de Béal et al. (1), les auteurs s'intéressent à la notion d'*équipe de gènes*. Cette notion est voisine de celle d'intervalle commun mais avec quelques contraintes relâchées. Par exemple, les doublons sont permis dans les équipes de gènes alors qu'ils ne le sont pas dans les intervalles communs. Biologiquement, les équipes de gènes sont plus réalistes que les intervalles communs.

## ANNEXE A

### LA DISTANCE D'INVERSION

Dans ce travail, nous avons mentionné, à maintes reprises, la notion de distance d'inversion d'une permutation signée. Dans cette annexe, nous essayerons de la présenter d'une façon très brève. Pour plus d'information à ce sujet, nous invitons le lecteur intéressé à consulter l'article de Bergeron et al. (4).

La définition d'une permutation signée qu'on trouve dans l'article de Bergeron et al. (4) est semblable à celle que nous donnons dans ce travail avec une légère différence.

En effet, les permutations utilisées préalablement dans ce travail sont basées sur l'ensemble  $\{1, 2, \dots, n\}$ , où  $n$  est le nombre d'éléments de la permutation. Dans l'article de Bergeron et al. (4), une permutation est basée sur l'ensemble  $\{0, 1, 2, \dots, n\}$  où  $n$  est le plus grand élément de cette permutation, de plus, elle commence toujours par l'élément 0 et se termine par le plus grand élément, soit  $n$ .

Dans cette annexe, nous allons modifier notre façon de noter une permutation en lui ajoutant l'élément 0, au début, et l'élément  $(n + 1)$ , à la fin. Ainsi nous obtenons la même notation que celle utilisée dans l'article de Bergeron et al. (4). Par exemple, la permutation  $(\bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2})$  devient  $(0 \ \bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2} \ 7)$ .

Du point de vue biologique, cette modification est cohérente car l'orientation d'un chromosome entier n'est pas pertinente. En effet, les seuls changements d'orientation intéressants sont ceux qui se passent à l'intérieur d'un chromosome.

Avant de discuter de la distance d'inversion d'une permutation comme telle, nous allons d'abord présenter certaines définitions nécessaires pour comprendre cette notion.

**Définition 16.** Dans une permutation signée  $P$ , un *point*  $p.q$  est défini par une paire d'éléments consécutifs dans  $P$ .

Par exemple, dans la permutation  $P = (0 \ \bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2} \ 7)$ , le point  $0 \ . \ \bar{4}$  est le premier point de  $P$  et  $\bar{2} \ . \ 7$  est son dernier point.

Dans n'importe quelle permutation signée  $P$  sur l'ensemble  $\{0, 1, 2, \dots, n\}$  qui commence par l'élément 0 et qui finit par l'élément  $n$ . Tout élément  $i$  de  $P$ , tel que  $(0 < i < n)$ , possède un point gauche et un point droit.

**Définition 17.** Une paire d'éléments non-signés  $(k, k+1)$  définit un *intervalle élémentaire*  $I_k$ , associé aux points  $k$  et  $k+1$ , dont les extrémités sont :

- 1) le point à droite de  $k$ , si  $k$  est positif, et le point à gauche sinon
- 2) le point à gauche de  $k+1$ , si  $k+1$  est positif, et le point à droite sinon.

Dans la Figure A.1, nous avons deux exemples d'intervalles élémentaires sur la permutation  $P = (0 \ \bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2} \ 7)$ , soit l'intervalle élémentaire  $I_0$  qui est associé aux éléments  $(0, 1)$  et l'intervalle élémentaire  $I_1$  associé aux éléments  $(1, 2)$ .

$$P = ( 0 \quad \bar{4} \quad 6 \quad 1 \quad \bar{3} \quad \bar{5} \quad \bar{2} \quad 7 )$$

**FIG. A.1** Exemples de deux intervalles élémentaires de la permutation  $P = (0 \ \bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2} \ 7)$ . L'intervalle élémentaire  $I_0$  est associé aux éléments  $(0, 1)$  et  $I_1$  est associé aux éléments  $(1, 2)$ .

**Définition 18.** Un *cycle* est une séquence de points  $b_1, b_2, \dots, b_k$  tels que deux points qui se suivent dans cette séquence sont des extrémités d'un intervalle élémentaire,  $b_1$  et  $b_k$  inclus.



Pour identifier les cycles d'une permutation, il suffit d'identifier tous les intervalles élémentaires de la permutation puis, constituer des cycles avec ces intervalles en respectant la contrainte donnée dans la Définition 18.

Pour tracer un cycle dans une permutation donnée  $P$ , on commence par un point donné de  $P$  et, en suivant les intervalles élémentaires, on parcourt un chemin unique qui revient à ce même point de départ.

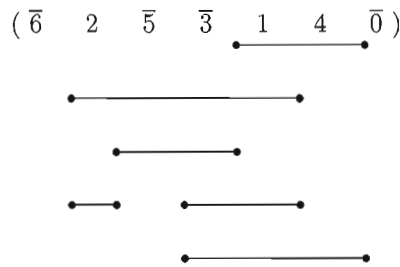


FIG. A.2 La trace qui montre les cycles de la permutation  $P = (2 \ 5 \ 3 \ 1 \ 4 \ 0)$

Dans la Figure A.2, nous montrons une trace qui identifie les cycles de la permutation  $P = (\bar{6} \ 2 \ \bar{5} \ \bar{3} \ 1 \ 4 \ \bar{0})$ . Il s'avère que  $P$  possède un seul cycle composé par les points suivants dans l'ordre :  $\bar{6} . 2 \rightarrow 1 . 4 \rightarrow \bar{5} . \bar{3} \rightarrow 4 . \bar{0} \rightarrow \bar{3} . 1 \rightarrow 2 . \bar{5} \rightarrow \bar{6} . 2$ . Notons que ce cycle pourrait commencer à n'importe quel point de  $P$ .

**Définition 19.** La *distance d'inversion* d'une permutation signée  $P$  est le nombre minimal d'inversions nécessaires pour transformer  $P$  en la permutation identité.

Dans l'article de Bergeron et al. (4), les auteurs utilisent une formule qui évalue une borne inférieure de la distance d'inversion d'une permutation signée  $P$  sur l'ensemble  $\{0, 1, 2, \dots, n\}$ , notée  $d(P)$ , sachant que le nombre de cycles dans la permutation  $P$  est égale à  $c$ . Cette formule est :

$$d(P) \geq n - c$$

Dans le dernier chapitre de ce travail, nous donnons des scénarios qui trient certaines permutations vers soit l'identité, soit l'identité inversée puis, nous affirmons qu'ils

sont parcimonieux. Seulement, nous n'expliquons pas comment nous savons que les longueurs de ces scénarios sont minimales. Dans la suite de cette section, nous allons justifier ces affirmations.

Soit une permutation signée  $P$  sur l'ensemble  $\{1, 2, \dots, n\}$ . Voici un algorithme qui calcule la borne inférieure de la distance d'inversion de  $P$  :

**Algorithme 6** : Calcul de la borne inférieure de la distance d'inversion d'une permutation.

**Entrée** : La permutation signée  $P$  sur l'ensemble  $\{1, 2, \dots, n\}$ .

**Sortie** : Un entier  $x$  tel que  $d(P) \geq x$ .

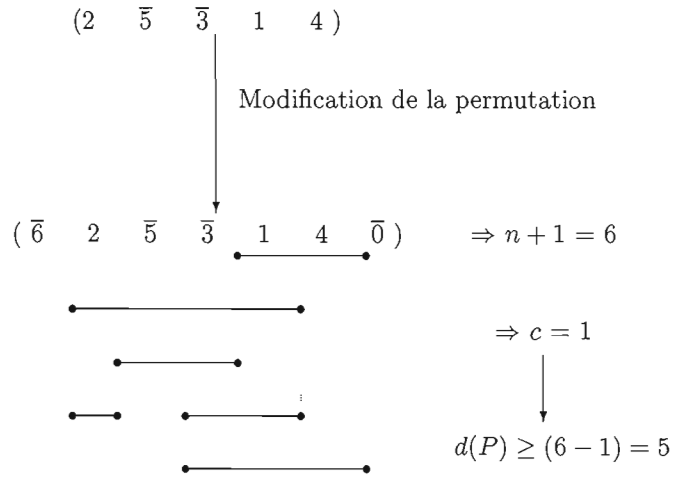
- 1 : ajouter à  $P$  les éléments aux extrémités : 0 au début, et  $(n + 1)$  à la fin;
- 2 : identifier les  $c$  cycles de la permutation résultante ;
- 3 :  $x = (n + 1) - c$ ,

### A.1 Le cas de la permutation $P = (2 \ \bar{5} \ \bar{3} \ 1 \ 4)$

Notons que le calcul de la distance d'inversion, tel que discuté dans l'article de Bergeron et al. (4) considère qu'on trie une permutation quelconque  $P$  vers la permutation cible qui est toujours l'identité. Or, dans notre travail, la permutation cible peut être soit  $Id$ , soit  $\overline{Id}$ .

La logique utilisée lorsque la permutation cible est l'identité inversée est semblable à celle où elle est l'identité. La différence se trouve au moment de la transformation de la permutation  $P$  sur l'ensemble  $\{1, 2, \dots, n\}$ .

En effet, lorsqu'on veut trier  $P$  vers  $Id_n$ , on lui ajoute 0, au début, et  $(n + 1)$  à la fin. Cependant, quand la cible est  $\overline{Id_n}$ , on ajoute à  $P$  l'élément  $\overline{(n + 1)}$ , au début, et  $\bar{0}$ , à la fin. Puis, nous identifions les cycles de  $P$  et nous calculons la borne inférieure de la distance d'inversion de  $P$  selon l'Algorithme 6.



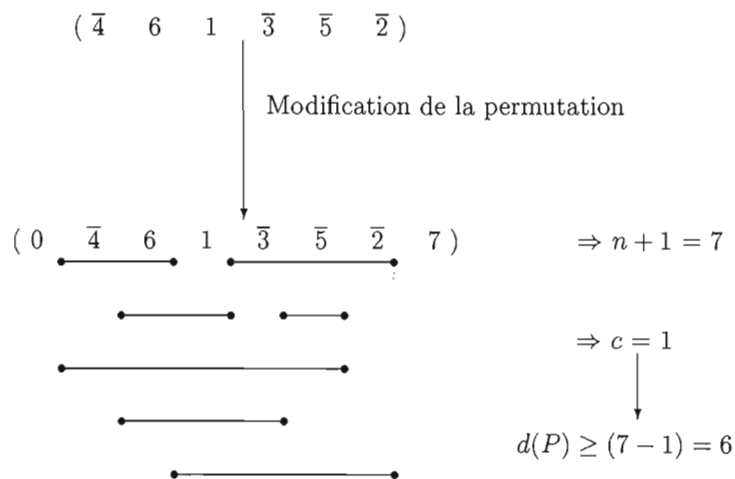
**FIG. A.3** Un scénario de longueur 5 qui trie  $P = (2 \bar{5} \bar{3} 1 4)$  vers  $\bar{Id}$  est parcimonieux car la longueur minimale d'une séquence d'inversions qui trie  $P$  vers  $\bar{Id}$  est 5.

Ainsi, dans la Section 3.2.2, nous avons une permutation  $P = (2 \bar{5} \bar{3} 1 4)$  que nous devons trier vers  $\bar{Id}$ . Nous affirmons, ensuite, que le scénario de la Figure 3.16 de longueur 5 qu'on trouve en utilisant un algorithme de tri de permutations signées est parcimonieux. En effet, comme on peut le voir dans la Figure A.3, une fois  $P$  transformée, son plus grand élément est égal à 6 et  $P$  contient 1 cycle, donc  $d(P) \geq 5$ . D'où, l'affirmation que le scénario de longueur 5 est parcimonieux.

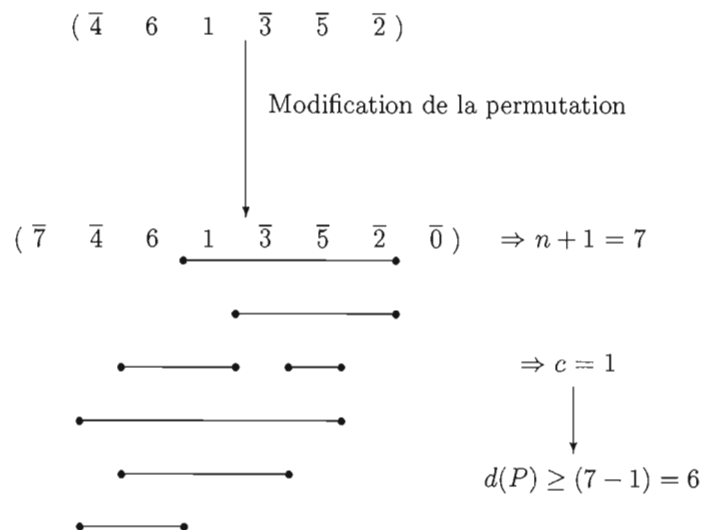
## A.2 Le cas de la permutation $P = (\bar{4} 6 1 \bar{3} \bar{5} \bar{2})$

Dans la Section 3.2.3, la permutation  $P = (\bar{4} 6 1 \bar{3} \bar{5} \bar{2})$  est de signe indéfini. Ainsi, nous devons essayer les deux sens et choisir le meilleur. Pour ce faire, nous calculons la borne inférieure de la longueur de la séquence qui trie  $P$  vers  $Id$  (voir la Figure A.4), puis celle de la longueur de la séquence qui trie  $P$  vers  $\bar{Id}$  (voir la Figure A.5).

Ensuite, on les compare et on réalise qu'elles ont la même valeur, soit 6. En utilisant l'algorithme de tri des permutations signées, on trouve deux scénarios de longueur 6 qui trient  $P$  l'une vers  $Id$  et l'autre vers  $\bar{Id}$ . Ainsi, on en conclut que les deux scénarios peuvent être des solutions et on en choisit un au hasard.



**FIG. A.4** Un scénario de longueur 6 qui trie  $P = (\bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2})$  vers  $Id$  est parcimonieux car la borne inférieure de la distance d'inversion de cette permutation est égale à 6 .



**FIG. A.5** Un scénario de longueur 6 qui trie  $P = (\bar{4} \ 6 \ 1 \ \bar{3} \ \bar{5} \ \bar{2})$  vers  $\overline{Id}$  est parcimonieux car la longueur minimale d'une séquence d'inversions qui trie  $P$  vers  $\overline{Id}$  est égale à 6.

## BIBLIOGRAPHIE

- (1) M.P. Béal, A. Bergeron, S. Corteel, et M. Raffinot. *An algorithmic view of gene teams*. Theoretical Computer Science, Volume 320(2-3), pages 395–418, 2004.
- (2) S. Bérard, A. Bergeron, C. Chauve, C. Paul. *Perfect Sorting by Reversals Is Not Always Difficult*. 5th Workshop on Algorithms in Bioinformatics (WABI 2005), Lecture Notes in Computer Science, Volume 3692, pages 228-237, Springer-Verlag, Berlin, 2005.
- (3) A. Bergeron, C. Chauve, F. de Montgolfier et M. Raffinot. *Computing common intervals of  $K$  permutations, with applications to modular decomposition of graphs*. Proceedings of 13th annual European Symposium on Algorithms (ESA 2005), Lecture Notes in Computer Science, Volume 3669, pages 779-790, Springer-Verlag, Berlin, 2005.
- (4) A. Bergeron, J. Mixtacki, et J. Stoye. *The inversion distance problem*. Mathematics of evolution and phylogeny, (O. Gascuel editor), pages 262-290, Oxford University Press, 2005.
- (5) M. Bernt, D. Merkle et M. Middendorf. *A Fast and Exact Algorithm for the Perfect Reversal Median Problem*. International Symposium on Bioinformatics Research and Applications (ISBRA), Lecture Notes in Computer Science, Volume 4463, pages 305-316, Springer-Verlag, Berlin, 2007.
- (6) S. Booth et S. Lueker. *Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms*. Journal of Computer and System Sciences, Volume 13, pages 335-379, 1976.
- (7) G. Bourque et P. A. Pevzner. *Genome-scale evolution : Reconstructing gene orders in the ancestral species*. Genome Research, Volume 12(1), pages 26–36, 2002.
- (8) G. Bourque, P. A. Pevzner et G. Tesler. *Reconstructing the genomic architecture of ancestral mammals : Lessons from human, mouse, and rat genomes*. Genome Research, Volume 14(4), pages 507–516, 2004.
- (9) A. Caprara. *Sorting by reversals is difficult*. Proceedings of the first annual international conference on Research in Computational Molecular Biology (RECOMB'97), (M. Waterman, editor), pages 75-83, 1997.
- (10) Y. Diekmann, M.-F. Sagot, et E. Tannier. *Evolution under Reversals : Parsimony and Conservation of Common Intervals*. IEEE/ACM Transactions on

- Computational Biology and Bioinformatics, Volume 4(2), pages 301-309, Oxford University Press, Oxford, 2007.
- (11) M. Figeac et J.-S. Varré. *Sorting by Reversals with Common Intervals*. Proceedings of the fourth International Workshop on Algorithms in Bioinformatics (WABI 2004). Lecture Notes in Computer Science, Volume 3240, pages 26-37, Springer-Verlag, Berlin, 2004.
  - (12) R. A. Gibbs et al. *Genome sequence of the brown norway rat yields insights into mammalian evolution*. Nature, Volume 428(6982), pages 493-521, 2004.
  - (13) M. T. Goodrich, R. Tamassia. *Data structures and algorithms in Java*. John Wiles and Sons, Inc., pages 157-162, 2004.
  - (14) S.Hannenhalli et P.A. Pevzner. *Transforming cabbage into turnip : Polynomial algorithm for sorting signed permutations by reversals*. Journal of the ACM (JACM), Volume 46(1), pages 1-27, New York, 1999.
  - (15) S. Heber et J. Stoye. *Finding all common intervals of k permutations*. In Proceedings of the 12th annual symposium on Combinatorial Pattern Matching (CPM 2001), Lecture Notes in Computer Science, Volume 2089, pages 207-218, Springer-Verlag, 2001.
  - (16) H. Ito, K. Iwama, et T. Tamura. Efficient methods for determining DNA probe orders. Special Section on Discrete Mathematics and Its Applications, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences , Volume E89-A,number 5, pages 1292-1298, 2006.
  - (17) G. M. Landau, L.Parida, et O. Weimann. *Using PQ Trees for Comparative Genomics*. In Proceedings of the 16th annual symposium on Combinatorial Pattern Matching (CPM 2005), Lecture Notes in Computer Science, Volume 3537, pages 128-143, Springer-Verlag, 2005.
  - (18) T. Marquès-Bonet, M. Cáceres, J. Bertranpetit, T. M. Preuss, J. W. Thomas et A. Navarro. *Chromosomal rearrangements and the genomic distribution of gene-expression divergence in humans and chimpanzees*. Trends in Genetics, Volume 20 (11), pages 524-529, November 2004.
  - (19) D. Sankoff. *Edit distance for genome comparison based on non-local operations*. In Proceedings of the 3rd annual symposium on Combinatorial Pattern Matching (CPM 1992), Lecture Notes in Computer Science, Volume 644, pages 121-135, Springer-Verlag, 1992.
  - (20) E. Tannier, A. Bergeron, et M.-F. Sagot. *Advances on Sorting by Reversals*. Discrete Applied Mathematics, Volume 155(6-7), pages 881-888, 2007.
  - (21) G. Tesler. *GRIMM : genome rearrangements web server*. Bioinformatics, Volume 18(3), pages 492-493, Oxford University Press, Oxford, 2002.

- (22) T. Uno et M. Yagiura. *Fast algorithms to enumerate all common intervals of two permutations*. Algorithmica, Volume 26(2), pages 290–309, 2000.
- (23) S. Yancopoulos, O. Attie, et S. Friedberg. *Efficient Sorting of Genomic Permutations by Translocation, Inversion and Block Interchange*. Bioinformatics, Volume 21(16), pages 3340–3346, Oxford University Press, Oxford, 2005.

## Code source Java

```
import ca.uqam.bioinfo.util.scenario.ReversalScenario;
import ca.uqam.bioinfo.util.permutation.Permutation;

public class ReversalScenarioMain {

    public ReversalScenarioMain() throws Exception {
        System.out.println("Test ReversalScenarioMain main : ");
        ReversalScenario rs = new ReversalScenario();
        int[] permTest = {-4,-3,-2,1,-13,-15,14,-16,8,9,10,-11,12,5,6,7};
        Permutation perm = new Permutation(permTest);
        rs.computePerfectScenario(perm);
        System.out.println("Fin test ");
        try {
            jbInit();
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {

        try {
            new ReversalScenarioMain();
        }
        catch(Exception e)
        {
            System.err.println("Exception in ReversalScenarioMain main : ");
            System.err.println(e.toString());
            e.printStackTrace();
        }

    }

    private void jbInit() throws Exception {
    }

}
```



```

package ca.uqam.bioinfo.util.scenario;

import ca.uqam.bioinfo.util.commoninterval.*;
import ca.uqam.bioinfo.util.permutation.PermutationSet;
import ca.uqam.bioinfo.util.pgtree.*;
import ca.uqam.bioinfo.util.permutation.Permutation;
import ca.uqam.bioinfo.util.interval.Interval;
import ca.uqam.bioinfo.util.distance.ReversalDistance;
import java.util.*;

public class ReversalScenario {

    private Vector<Permutation> permutationScenario = new Vector<Permutation>
();
    private Vector<PQNode> signedPNodes = new Vector<PQNode> ();
    private Vector<PQNode> undefPNodes = new Vector<PQNode> ();
    private Vector<Interval> intervalScenario = new Vector<Interval> ();
    private int undefNode=0;
    private int primarNode=0;

    /** Returns the intervals scenario from the source permutation to the
destination permutation.
    * This scenario contains reversals on all types of nodes (leaves, qNodes
and pNodes)
    * @return Vector
    */
    public Vector<Interval> getIntervalScenario() {
        return this.intervalScenario;
    }

    public Vector<Permutation> getPermutationScenario() {
        return this.permutationScenario;
    }

    /**Adds the given interval to the scenario under intervals form from the
source permutation
    * to the destination permutation. Only reversals on leaves and linear
nodes.*/
    public void addToIntervalScenario(Interval interv) {
        intervalScenario.add(interv);
    }
}

```

```

/** @description Traverses the tree from the root pq, assigning signs to every
node of the tree*/
public void addSign(PQNode pq, Permutation p) throws Exception {
    PQNode temp;
    if (pq.isLeaf()) {
        if (p.inverse().getElement(pq.getValue())<0)
            pq.setSign("-");
        else
            pq.setSign("+");
    }
    else if (pq.isQNode()) {
        TreeMap<Integer, Integer> tm;
        tm= getChildrenPositions(p,pq);
        if(tm.get(tm.firstKey())<tm.get(tm.lastKey())){
            pq.setSign("+");
        }else {pq.setSign("-");}
    }
    else if (pq.getParent()==null || pq.getParent().isPNode()) {
        pq.setSign("undef");
    }
    else
        pq.setSign(pq.getParent().getSign());

    if (!pq.isLeaf()) {
        for (int i = 0; i < pq.getChildren().length; i++) {
            temp = (PQNode) pq.getChildren()[i];
            addSign(temp, p);
        }
    }
}

public void identifyTypeTree(PQNode pq) throws Exception {
    PQNode temp;
    if ((pq.isPNode()) && (pq.getSign()=="undef")){
        this.undefPNodes.add(pq);
        undefNode++;
    } else if (pq.isPNode()==true){
        this.signedPNodes.add(pq);
        primarNode++;
    }
    if (!pq.isLeaf()) {
        for (int i = 0; i < pq.getChildren().length; i++) {
            temp = (PQNode) pq.getChildren()[i];
            identifyTypeTree(temp);
        }
    }
}

/** Assigns a sign to the root of PQTree, then its subtrees*/
public void addParent(PQNode pq) {
    pq.setParent(null);
    addParentRec(pq);
}

```

```

/** Traverses the subtrees of a node pq (the root of the tree), assigning
parent to every node */
public void addParentRec(PQNode pq) {
    PQNode temp;
    if (pq != null && (!pq.isLeaf())) {
        for (int i = 0; i < pq.getChildren().length; i++) {
            temp = (PQNode) pq.getChildren()[i];
            temp.setParent(pq);
            if (!temp.isLeaf()) { addParentRec(temp); }
        }
    }
}

public TreeMap<Integer, Integer> getChildrenPositions(Permutation p, PQNode
pq) throws Exception {

    int[] children = new int[pq.getChildren().length];
    children = getLastElementOfChildren(pq);
    int[] childrenPos = new int[children.length];
    for (int i = 0; i < childrenPos.length; i++) {
        childrenPos[i] = Math.abs(p.inverse().getElement(Math.abs(children[i])));
    }
    TreeMap<Integer, Integer> tm = new TreeMap<Integer, Integer> ();
    for (int i = 0; i < children.length; i++) {
        tm.put(childrenPos[i], children[i]);
    }
    return tm;
}

/** Returns a quotient permutation of a node pq from a tree made with the
permutation perm */
public Permutation computePermQuotien(PQNode pq, Permutation p) throws
Exception {
    int[] children = new int[pq.getChildren().length];
    children = getLastElementOfChildren(pq);
    int[] childrentem = new int[children.length];
    for (int i = 0; i < childrentem.length; i++) {childrentem[i]=i; }
    TreeMap<Integer, Integer> tm;
    tm= getChildrenPositions(p,pq);
    TreeMap<Integer, Integer> tm2 = new TreeMap<Integer, Integer>();
    for (int i = 0; i < children.length; i++) {
        tm2.put(children[i],i+1);
    }
    int[] tab2 = new int[tm.size()];
    int i=0;
    for (Iterator it=tm.keySet().iterator(); it.hasNext(); ) {
        int value = tm.get((Integer) it.next());
        if (value<0) {tab2[i]=-(tm2.get(value));} else {tab2[i]=(tm2.get(value));}
        i++;
    }
    Permutation PermQuot = new Permutation(tab2);
    return PermQuot;
}

```

```

/** Returns an array that contains the last element (int)of each child of a
given node pq. */
public static int[] getLastElementOfChildren(PQNode pq) {
    int[] children = new int[pq.getChildren().length];
    PQNode temp;
    if (pq != null && !pq.isLeaf()) {
        for (int i = 0; i <= pq.getChildren().length - 1; i++) {
            temp = (PQNode) pq.getChildren()[i];
            if (temp.getSign() == "-") { children[i] = -temp.getValue(); }
            else { children[i] = temp.getValue(); }
        }
    }
    return children;
}

public void computePerfectScenario(Permutation p) throws Exception {
    PermutationSet pSet = new PermutationSet();
    pSet.add(p);
    pSet.add(new Permutation(p.Lenght()));
    CanonicalGenerator cgen = new CanonicalGenerator(pSet);
    PQNode pq = cgen.computePQTree();
    addParent(pq);
    addSign(pq, p);
    identifyTypeTree(pq);
    System.out.println("undef= "+this.undefNode+" primarNode="+this.primarNode);
    if(this.undefNode>0) {
        treatAmbiguousTree(p, pq);
    } else if (this.primarNode>0){
        treatNonAmbiguousTree(p, pq);
    } else {
        treatDefiniTree(p,pq);
    }
    for (int i=0; i<=this.permutationScenario.size()-1; i++){
        System.out.println(this.permutationScenario.elementAt(i));
    }
    System.out.println(this.intervalScenario.toString());
}

public void treatAmbiguousTree(Permutation p, PQNode pq) throws Exception {
    System.out.println("L'arbre est ambigu");
    for (int i = 0; i <= this.undefPNodes.size() - 1; i++) {
        System.out.println( ("nonSignes: "+ i+" "+
this.undefPNodes.elementAt(i)).toString());
    }
    treatNonAmbiguousTree(p,pq);
}

public void treatNonAmbiguousTree(Permutation p, PQNode pq) throws Exception {
    treatDefiniTree(p, pq);
    treatNonAmbiguousTreeRec(p,pq);
}

```

```

public void treatNonAmbiguousTreeRec(Permutation p, PQNode pq) throws Exception
{
    Vector<int[]> scenPQuot;
    PQNode temp;
    Permutation pQuot;
    if (pq.getSign() == "+") {
        pQuot = computePermQuotien(pq, p);
        scenPQuot = ReversalDistance.sort(pQuot, this.intervalScenario);
        int[] inv = scenPQuot.get(0);
        Interval interv = transformIdInvScen(pq, pQuot, p, inv);
        this.intervalScenario.add(interv);
        applyInv(p, interv);
        this.permutationScenario.add(p.copy());
    }
    for (int i = 0; i <= this.signedPNodes.size() - 1; i++) {
        temp = this.signedPNodes.elementAt(i);
        pQuot = computePermQuotien(temp, p);
        scenPQuot = ReversalDistance.sortToIdReversed(pQuot,
            this.intervalScenario);
        for (int j = 0; j <= scenPQuot.size() - 1; j++) {
            int[] inv = scenPQuot.get(j);
            Interval interv = transformIdInvScen(temp, pQuot, p, inv);
            this.intervalScenario.add(interv);
            applyInv(p, interv);
            this.permutationScenario.add(p.copy());
        }
    }
}

public void treatDefiniTree(Permutation p, PQNode pq) throws Exception {
    PQNode temp;
    Interval interv = new Interval();
    if (pq.getParent() != null) {
        if ( (pq.getSign() != (pq.getParent()).getSign()) &&
            (pq.getParent().isQNode()) ) {
            if (pq.isLeaf()) {
                interv = new Interval(pq.getValue(), pq.getValue());
            }
            else if (pq.isQNode()) {
                interv = pq.getEquivalentInterval();
            }
            this.intervalScenario.add(interv);
            applyInv(p, interv);
            this.permutationScenario.add(p.copy());
        }
    }
    if (!pq.isLeaf()) {
        for (int i = 0; i <= pq.getChildren().length - 1; i++) {
            temp = (PQNode) pq.getChildren()[i];
            treatDefiniTree(p, temp);
        }
    }
}

```

```

public Interval transformIdInvScen(PQNode pnode, Permutation permQuot,
                                   Permutation p, int[] inv) throws Exception
{
    Interval interInv = new Interval();
    PQNode premierElt, dernierElt;
    int indice = 0;
    PQNode[] children = new PQNode[pnode.getChildren().length];
    for (int i = 0; i <= children.length - 1; i++) {
        children[i] = (PQNode) pnode.getChildren()[i];
    }
    TreeMap<Integer, PQNode> tm2 = new TreeMap<Integer, PQNode> ();
    for (int i = 0; i < children.length; i++) {
        tm2.put(i + 1, children[i]);
    }
    premierElt = tm2.get(Math.abs(inv[0]));
    TreeMap<Integer, Integer> tmPremier = new TreeMap<Integer, Integer> ();
    Interval interInvP = premierElt.getEquivalentInterval();
    if (interInvP.getR() > interInvP.getL()) {
        for (int i = interInvP.getL(); i <= interInvP.getR(); i++) {
            tmPremier.put(p.inverse().getUnsignedElement(i), i);
        }
    }
    else {
        for (int i = interInvP.getR(); i <= interInvP.getL(); i++) {
            tmPremier.put(p.inverse().getUnsignedElement(i), i);
        }
    }
    for (int k = 0; k <= inv.length - 1; k++) {
        if (inv[k] != 0) {
            indice++;
        }
    }
    dernierElt = tm2.get(Math.abs(inv[indice-1]));
    Interval interInvD = dernierElt.getEquivalentInterval();
    TreeMap<Integer, Integer> tmDernier = new TreeMap<Integer, Integer> ();
    for (int i = interInvD.getL(); i <= interInvD.getR(); i++) {
        tmDernier.put(p.inverse().getUnsignedElement(i), i);
    }
    interInv.setL(tmPremier.firstKey());
    interInv.setR(tmDernier.lastKey());
    return interInv;
}

public void applyInv(Permutation p, Interval interv) throws Exception {
    Interval interv2 = new Interval();
    interv2.setL(interv.getL() - 1);
    interv2.setR(interv.getR() - 1);
    p.reverseInterval(interv2);
}

```

```

public int[][] makeSignMatrix(int k) {
    int rowCount = (int) Math.pow(2, k);
    int[][] matrix = new int[rowCount][k];
    int step = 1, row;

    for (int col = k-1; col >= 0; col--) {
        row = 0;
        while (row < rowCount) {
            for (int j = 0; j < step; j++) {
                matrix[row][col] = 0;
                row++;
            }
            for (int j = 0; j < step; j++) {
                matrix[row][col] = 1;
                row++;
            }
            step *= 2;
        }

        return matrix;
    }
}

```